

MAGAZINE MSX
AÑO II
Núm. 16
Septiembre
1986
300 Ptas.
DOS GIGANTES FRENTE A FRENTE
TEST: VC-10, un osciloscópio muy especial
HIT BIT
Síntesis de voz
Libros, programas, software
Utilidades de la RAM

DIRECTOR:
Juan Arencibia.

COLABORADORES:
Octavio López, Angel Zarazaga, Teresa Aranda, Ricardo García.

DISEÑO:
Benito Gil

Editada por:
PUBLINFORMATICA, S.A.
C/ Bravo Murillo, 377 - 5.º A
Tel.: 733 74 13
28020 Madrid.
Telex 48877 OPZXE

PRESIDENTE:
Fernando Bolín.

DIRECTOR EDITORIAL REVISTAS DE USUARIOS:
Juan Arencibia.

DIRECTOR DE VENTAS:
Antonio González.

JEFE DE PRODUCCIÓN:
Miguel Onieva.

SERVICIO AL CLIENTE:
Julia González.
Tel.: 733 79 69

DIRECCION, REDACCION Y ADMINISTRACIÓN:
C/ Bravo Murillo, 377 - 5.º A.
Tel.: 733 74 13
28020 Madrid.

PUBLICIDAD EN MADRID:
Emilio García.

PUBLICIDAD EN BARCELONA:
Lidia Cendros.
C/ Pelayo, 12.
Tel.: (93) 301 47 00 Ext. 27-28.
08001 Barcelona.

Depósito Legal: M. 16.755-1985
Impreso en Héroes, S.A.
C/ Torrelara, 8. 28016 Madrid.
Distribuye:
S.G.E.L. Avda. Valdelaparra, s/n.
Alcobendas (Madrid).

DISTRIBUIDORES:
VENEZUELA: SIPAM, S.A.
Avda. República Dominicana, 541
ARGENTINA: DISTRIBUIDORA INTERCONTINENTAL BUENOS AIRES.

El P.V.P. para Ceuta, Melilla y Canarias, incluido servicio aéreo será de 300 ptas. sin I.V.A.

SUSCRIPCIONES
Rogamos dirija toda la correspondencia relacionada con suscripciones a:
MSX
EDISA: Tel, 415 97 12
C/López de Hoyos, 141-5.º
28002 MADRID
(Para todos los pagos reseñar solamente MSX)
Para la compra de ejemplares atrasados diríjanse a la propia editorial
MSX
C/Bravo Murillo, 377-5.º A
Tel. 733 74 13 28020 MADRID

Si deseas colaborar en MSX remite tus artículos o programas a Bravo Murillo 377, 5.º A. 28020 Madrid. Los programas deberán estar grabados en cassette y los artículos mecanografiados.
A efectos de remuneración, se analiza cada colaboración aisladamente, estudiando su complejidad y calidad.



A un paso de finalizar el verano, nos hemos visto sorprendidos por la aparición de dos nuevos ordenadores de la II generación, ambos de la misma casa. Efectivamente, Mistubishi destapó el tarro de las esencias y lanzó al mercado dos ordenadores que van a marcar una pauta difícil de igualar.

Ni Sony, con su HB-500P, ni Philips con el VG-8235, van a ser rivales de importancia cuando Mabel (distribuidor de Mitsubishi en nuestro país) ataque seriamente con sus productos. ML-G1 y ML-G3 son las denominaciones de estos aparatos, cuyas características resaltaremos en un completo estudio compartivo que efectuaremos para el mes de octubre. Esto relanzará el estándar y recrudecerá la lucha en el mercado de los ordenadores de la II generación. Sólo el tiempo y la aceptación por parte del público, podrán predecir el resultado.

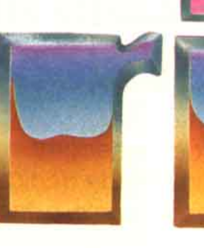
Por otro lado, este mes realizamos una comparativa de excepción, HB-500P contra VG-8235. Ambos ordenadores tienen su propio estilo y, si por un lado nos gustó el Philips por su versatilidad, por parte de Sony cabe destacar su aspecto más profesional. De cualquier modo, el futuro de ambos aparatos se encuentra en la compatibilidad y programas que puedan ofrecer. Hasta la fecha, existen pocos programas de aplicación para estos ordenadores, pero las casas comerciales están trabajando en ello.

Además de esta comparativa, probamos un periférico que convierte el ordenador en un osciloscopio; el NEOS VC-10. También analizamos en un artículo como se puede realizar un sintetizador con el MSX y la mejor forma de aprovechar la memoria del ordenador.




HIT BIT
MSX 2
F500P





NOTICIAS

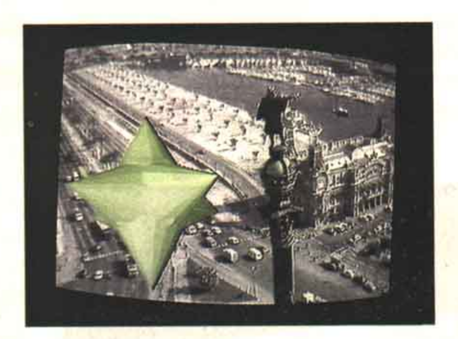
Estuvimos en Videografía

Videografía es una institución privada dedicada única y exclusivamente a las actividades pedagógicas, culturales y artísticas del vídeo. Escuela, productora y distribuidora de vídeo experimental y grafismo electrónico, comenzó sus actividades en 1981, manteniendo una presencia activa en el sector de la comunicación audiovisual en nuestro país.

Imparte cursos de Formación técnico-profesional, realización de vídeo informativo y documental, especiales de vídeo-art y vídeo creación, monográficos de iluminación, sistemas, señal de vídeo y ordenador, además de los cursos de introducción al grafismo digital e informatización del vídeo en los que estuvimos presentes.

Los trabajos se realizaron con ordenadores MSX de Sony, un Gen Locker, el nuevo MSX-2, HB-G900P con Videotizer y los monitores y aparatos de vídeo necesarios. El curso, después de una introducción sobre la informatización del vídeo, se basó principalmente, en la generación y creación de caracteres y gráficos digitales, digitalización de imágenes de vídeo y ejercicios prácticos de grafismo digital.

De la misma manera que las formas tradicionales de expresión



gráfica han conseguido un desarrollo personal, el diseño gráfico avanza marcando unas pautas propias en el arte de la imagen digital.

Para obtener más información, dirigirse a:

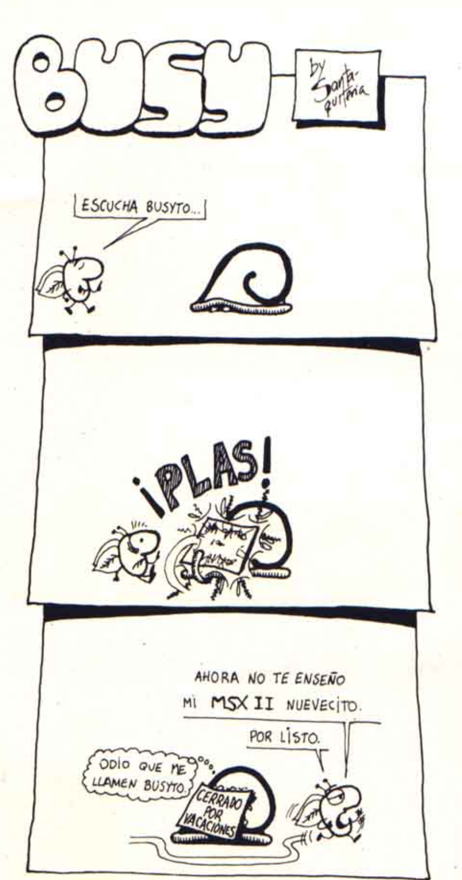
Videografía
Berlín, 72
Barcelona 08029

Konami Shop, una tienda especial

Durante el pasado mes de agosto, SERMA inauguró en la calle Francisco Navacerrada, 19 (Madrid) una tienda dedicada a la comercialización y distribución de los juegos de Konami.

Además y aprovechando el descanso veraniego, la conocida empresa cambió de aires. Efectivamente, el domicilio actual es calle Cardenal Belluga, 21 (Madrid). En ella, continuarán distribuyendo todo tipo de programas, especialmente los de Konami, fabricante japonés cuya experiencia en nuestro país está siendo muy favorable.

La novedad más interesante viene de Inglaterra, donde la aparición de un strip poker que ha hecho furor en otros ordenadores, tiene su más claro exponente en la adaptación de este juego para ordenadores MSX. Samantha Fox, va a ser sin lugar a dudas, lo más novedoso de la época estival.



Mitsubishi, Primer fabricante con dos ordenadores de la 2.ª generación

La aparición de los ordenadores MSX II sigue siendo novedad. Actualmente, Mitsubishi con dos modelos se ha situado en la cresta de la ola, presentando el ML-G1 y el ML-G3. Ambos ordenadores tienen sus propias características pero son de similar concepción. Sin duda alguna, se convertirán (sobre todo el modelo ML-G3) en una seria amenaza para ordenadores de la talla de Sony y Philips. Por lo pronto y para adelantar algunas características, diremos que es el primero que lleva la tecla «ñ» incorporada directamente al teclado. Con esto se evita tener que introducirla mediante las consabidas teclas «SHIFT» + «n». La compatibilidad parece estar asegurada, aunque esto lo veremos con más detenimiento en la prueba que haremos para el mes de octubre, donde analizaremos ambos ordenadores y una unidad de discos para el ML-G1.

Más libros de Anaya

«Biblioteca de Macros para Lotus 1-2-3» y «dBIII. Guía avanzada del programador», son dos de los libros que aparecerán este verano.

El primero de ellos, de 324 páginas, con un precio de 5.300 ptas. (con un diskette incorporado), describe con ejemplos concretos la construcción de macros para automatizar multitud de tareas frecuentes con LOTUS. Mediante la información del libro, se puede incrementar en diez veces, la velocidad de funcionamiento del programa. El diskette que incorpora el libro, tiene grabado todas las macros.

«dBIII. Guía del programador» enseña al lector los principios de gestión de Bases de Datos, la utilización correcta de las instrucciones y los métodos para diseñar sus propios y sofisticados sistemas de Bases de Datos. El precio, IVA incluido, es de 3.339 ptas. y tiene 506 págs.

Interface Midi y Software para ordenadores MSX

ELECTROMUSIC MIDI y SOFTWARE acaba de lanzar un paquete compuesto por un Interface MIDI para los ordenadores MSX, con las siguientes prestaciones y con el programa en disco o cassette.

El programa permite controlar hasta 8 instrumentos musicales electrónicos (sintetizadores, órganos, baterías, etc.) con entradas y salidas MIDI, y permite la utilización del ordenador como secuenciador de 8 pistas completamente polifónicas aunque, utilizando la función Track Merge, es posible hacerlo con hasta 29 pistas.

Otra de las posibilidades es la denominada Loop, que permite la reproducción repetitiva de determinadas pistas en sincronismo con otras no repetitivas e incluso de distinta duración. En tal caso, el ajuste entre ambas se realiza de modo completamente automático.

El programa ofrece en pantalla lectura de la cantidad de memoria que resta libre. La capacidad indicada en las especificaciones es de 32.348 eventos, lo que equivale aproximadamente a 6.500 notas.

El importador de estos productos es:

Sistema Midi
Córcega, 89, entlo.
Barcelona 08029



VG-8235 contr

El mercado de los ordenadores de la 2.ª Generación, aunque todavía es nuevo, tiene en estos dos aparatos sus mejores armas. Son idénticos en prestaciones, pero diferentes en concepción.

Desde que se anunciara la aparición de los ordenadores de la 2.ª Generación, se han vertido ríos de tinta sobre el tema, hasta su posterior comercialización en nuestro país. Previamente, estos ordenadores han dado la campanada en otros países



n HB-F500P



europeos, convirtiéndose (en el caso del Philips) en auténticas herramientas didácticas empleadas en aulas informáticas de colegios holandeses.

Por nuestro lado, hemos dedicado un número de la revista a explicar las características de esta generación (núm. 10 del mes de febrero) y otro al primero que se lanzó al mercado español, el Philips VG-8235 (comentado en el núm. 13 del mes de mayo), por esta razónn omitiremos algunos de los comentarios realizados en aquellos momentos y que hacen referencia a dichas cualidades, como por ejemplo, los diversos modos de pantalla, 512 colores, disco *RAM,* etc. En los ordenadores que vamos a comentar, estas características van ligadas directamente a ellos y no se pueden considerar como elementos que los diferencien.

Sin embargo, estos ordenadores se han presentado en un momento clave del mercado español, cuando ya existen aparatos cuya competencia se presentará muy dura. Por un lado tenemos al Amstrad 6128, Commodore C-128 y el nuevo Sinclair, y por otra parte, aunque no pueda competir debido al precio (unas 350.000 ptas.) encontramos el Commodore Amiga. Todos tienen buenas prestaciones y, sin embargo, la competencia más fuerte vendrá desde dentro, de los mismos fabricantes de MSX. Entre ellos se establecerá una importante batalla que se decantará a favor del ordenador no sólo más completo, sino del más práctico.

Dos planteamientos distintos

Cuando vemos el Son y HB-F500P por primera vez, nos choca su concepción. Todo es diferente, lo que da una sensación de estar frente a un ordenador que deja de ser personal para convertirse en semi profesional. Quizá se deba a su teclado independiente (con un total de 90 teclas) o a su teclado numérico (situado a la derecha del tecladod principal), pero en realidad estamos frente a un ordenador personal concebido para tareas más importantes y útiles que el juego. Teclado y unidad central forman un conjunto armonioso y elegante que unido al color (blanco) forma un bloque homogéneo.

Sin embargo, el VG-8235, es el otro lado de la moneda. Se trata de



un ordenador nada ostentoso, de un color gris oscuro y teclado sencillo (compuesto de 73 teclas), aunque muy práctico. En ambos ordenadores se puede conseguir, aunque de manera distinta, una cómoda inclinación del teclado para facilitar la introducción de datos. En el Philips, se consigue soltando dos pestañas que permiten levantar el teclado (consiguiendo un total de 3 ángulos distintos), mientras que en Sony se efectúa el desplegar dos patas de la parte inferior. De cualquiera de las dos formas se logra una posición óptima en cada momento. La unidad de discos de 3.5 pulgadas es una característica que poseen ambos ordenadores. Sin embargo, mientras que HB-F500P permite formatearlos a simple y doble carga, el VG-8235 sólo lo hace a una cara. De esta manera, en el modelo de Sony se consiguen obtener de cada disco hasta 720K, el doble que la unidad de discos de Philips. Este puede ser uno de los puntos más importantes que puede decantar el resultado de uno frente al otro, pero no es determinante. Y llegamos a los conectores e interfaces.

Dos ordenadores que dejan la categoria de personales para subir a la de semiprofesionales.

Aquí se puede decir que Sony no ha escatimado medios a la hora de poner buses de expansión, pero habría que ver hasta qué punto es realmente útil poseer 3 conectores para cartuchos *ROM,* además del conector para una unidad de discos. Philips se muestra fiel a la marca y mantiene los dos buses, aunque ahora existe un interface para una unidad de disco, igual que en el Sony. Otro punto que tienen ambos ordenadores en común es que poseen una tecla *RESET,* la del Philips más accesible que la del Sony, pero de nuevo, es cuestión de adaptarse.

En cuanto a interfaces, hay que resaltar un punto sumamente importante que nos ha sorprendido bastante (negativamente), el HB-F500P no posee conector de antena para TV.

Efectivamente, que un ordenador con un precio que ronda las 140.000 ptas. no tenga este importante conector nos parece poco serio, sobretodo cuando la

Pros del VG-8235

- Angulos de inclinación del teclado.
- Completa gama de conectores e interfaces.
- Discos con el MS-DOS, base de datos, tratamiento de textos y diseñador.
- Precio más asequible.

Contras del VG-8235

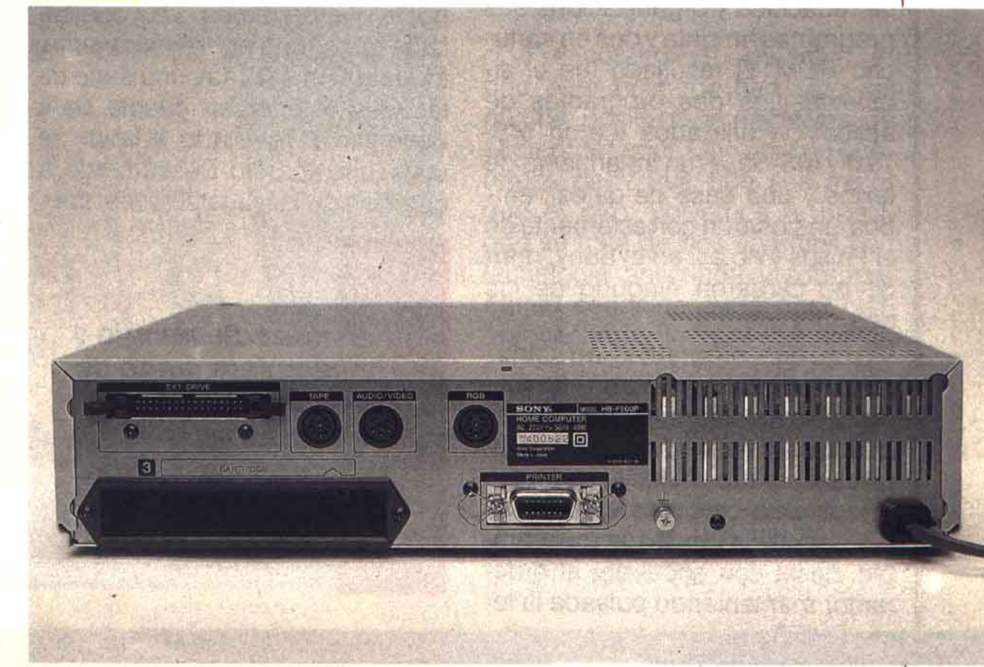
- Sólo permite utilizar discos de simple cara.
- Falta un teclado numérico.
- Muy voluminoso.

mayoría de los usuarios trabajan contínuamente con la «tele». Este punto, obligará al usuario potencial a comprarse un monitor.

Indudablemente queda mucho mejor y es más bonito trabajar con monitor (a ser posible en color) antes que con una TV, pero ¿vale realmente la pena gastarse un dinero adicional en la compra de un monitor sólo por tener este conector? Aquí sí que podemos decir que Philips no ha escatimado ni medios ni esfuerzos, ya que no sólo posee el conector para la antena de TV, sino que también cuenta con una salida para monitor *RGB* y el Euroconector, lo que da muestras de la versatilidad del equipo. Este último es muy útil para determinados monitores y TV, y para ciertas aplicaciones. Salvando esta gran diferencia (que sí puede ser determinante a la hora de escoger), ambos aparatos poseen los mismos interfaces, pero ubicados en zonas distintas. En el Sony, los dos *ports* para *joysticks* están situados en la parte frontal de la unidad central, mientraas que Philips lo tiene dispuesto al lado de la unidad de disco. En ambos modelos, tanto el interface Centronics como el del cassette, se encuentran en la parte posterior.

Podemos decir, sin temor a equivocarnos, que en el apartado de conectores e interfaces. Philips se lleva la placa. Viene preparado para poderlo utilizar con cualquier tipo de periférico existente y, como hemos dicho anteriormente, se muestra mucho más versátil.

Su presentación y el teclado independiente son las mejores armas del Sony.



Pros del HB-F500P

- Teclado independiente del ordenador.
- Teclado numérico con las operaciones aritméticas.
- Angulo de inclinación.
- Presentación del ordenador.
- Facilidad para formatear discos a simple y doble cara.

Contras del HB-F500P

- No tiene conector para antena de TV.
- No tiene Euroconector.
- Incompatible con algunos programas.
- Muy caro para la tecnología que emplea.

Software, el punto de la discordia

Aunque la política de compatibilidad tiende a ser generalizada entre los mismos fabricantes de MSX, hemos podido constatar una serie de pequeños fallos que complican esta importante característica, supuestamente la piedra angular de la filosofía MSX. La compatibilidad ha sido la clave principal de estos ordenadores y, a su vez, una arma de doble filo. Poder intercambiar periféricos y/o programas, era algo impensable hasta hace unos meses, pero gracias a estos ordenadores ya es posible. Sin embargo, la aparición de la 2.ª

Generacion, ha trastocado el mercado del *software*. En teoria, los programas gozan de compatibilidad ascendente. Esto quiere decir que todo el *software* existente en el mercado para los primeros ordenadores son compatibles con la 2.ª Generación de MSX. Puestos a probar, escogimos dos programas de aplicación en disco (muy utilizadas por nosotros en tareas de redacción y organización), dos programas en cinta y dos en cartucho *ROM*. El resultado fue el siguiente: Los dos programas de aplicación utilizados fueron *MS-Text* y *MS-Base* (un tratamiento de textos y una base de datos), ambos funcionaron correctamente en el Philips y no así en el Sony, donde no cargaron ninguno de los dos.

Los dos programas en cartuchos (juegos de *Konami*) funcionaron correctamente en ambos aparatos, y para los programas en cinta hubo que anular la unidad de discos para cargarlos (el proceso para anular la unidad es mu y simple, basta con encender el ordenador manteniendo pulsada la tecla *SHIFT*).

Esto nos' lleva a plantear una pregunta que se contestará con el tiempo. ¿Será compatible al 100% el *software* existente con estos ordenadores? En vista de estos resultados, el Philips tiende a ser más compatible que Sony, pero ¿hasta qué punto? Los programas continúan apareciendo, aunque ya hay más orientados a estos ordenadores específicos. Para ello, varias han sido las empresas que se están dedicando al desarrollo de programas para ordenadores de la 2.ª Generación. De estas podemos destacar a Idealogic, que en breve espacio de tiempo comercializará un *toolkit* para programadores.

De todos modos, aún es pronto para evaluar la penetración que han tenido estos ordenadores en nuestro país, ni sabemos a qué nivel de desarrollo se encuentran los programas de estos ordenadores.

Conclusión

Es difícil establecer una norma por la que podamos decir que éste es mejor que aquél y viceversa. Indudablemente, hay una serie de factores a tener en cuenta para decantar el resultado a favor de uno u otro aparato. Descartando la cuestión de las prestaciones, don-

Sencillez de manejo y un completo juego de conectores e interfaces, es la respuesta del Philips.



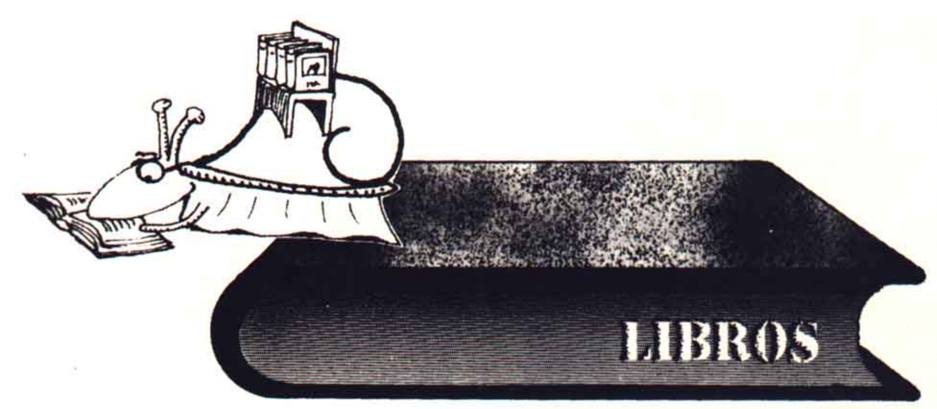


de no hay ganador alguno puesto que ambos están basados en tecnologías archiconocidas (un *Z-80A*, un procesador de vídeo *V9938*, etc.) habría que empezar por el precio. El VG-8235 cuesta unas 20.000 ptas. menos que el HB-F500P, lo cual es un aliciente para la mayoría de los usuarios, pero a su vez se muestra más discreto y no es tan imponente como el Sony.

El cuadro 1, muestra lo que para nosotros han sido los puntos positivos y negativos de cada ordenador. Con esto no queremos decir que uno sea mejor que el otro. Volvemos a insistir, que se trata de dos aparatos con tecnología anticuada y sobradamente conocida, pero muy bien aprovechada, lo que dificulta la elección de cualquiera de los dos.

Sin embargo, pueden influir los distintos puntos que hemos ido tratando a lo largo del artículo, tales como el precio, el conector de la antena de TV, la compatibilidad de un equipo, etc.





Libro: MSX de la A a la Z
Editorial: Ra-Ma
Páginas: 233

Con la aparición del sistema operativo MSX se produjo un hito histórico en el desarrollo de los microordenadores domésticos. Por primera vez, una serie de compañías de renombrado prestigio se ponían de acuerdo para fabricar sus aparatos bajo unas condiciones estándar.

Estas condiciones determinan, de hecho, la posibilidad de intercambio de programas y periféricos, es decir, la compatibilidad a todos los niveles, entre los distintos fabricantes.

Las especificaciones se basan en un microprocesador muy conocido, el *Z-80* en su versión A, que permite una mayor velocidad de funcionamiento, un procesador de vídeo *(VDP)* que dispone de memoria de vídeo independiente, un generador de sonido *(PSG) AY-3-8910,* con posibilidad de generación de tres canales de sonido diferentes, en tres octavas y un interfaz para periféricos *(PPI)* 8255 para el control del teclado, cassette, etc.

El lenguaje utilizado es el *MSX-BASIC,* una versión de *Microsoft,* que ocupa junto con el Sistema Operativo un total de *32K* en *ROM,* más un área de trabajo de *3K,* lo que deja unos *29K* libres para usuario, del total de *64K* direccionables simultáneamente por el microprocesador.

Esta versión es especialmente potente, al disponer además de las instrucciones usuales, de otras para tratamiento de archivos, control de sprites, trazado de gráficos mediante un microlenguaje especial, composición musical, etc., siendo semejante al de otros ordenadores de mayor tamaño.

El sistema también admite la utilización de discos flexibles, para lo que dispone de un sistema operativo de disco propio, el *MSX-DOS,* en disco o en cartucho, que completa el sistema operativo residente en *ROM* con todas las operaciones de disco. Así mismo, existe una versión de *BASIC* para disco.



El estándar admite también otros sistemas operativos, como el *CP/M* en su versión 2.2.

La colección de *Microguías,* en este volumen dedicado al sistema MSX, recoje todas aquellas expresiones relacionadas con el mismo, especialmente las instrucciones, funciones y operaciones del lenguaje *BASIC* y del lenguaje máquina *Z-80.*
Como promicia se esbozan las características de la próxima generación de equipos MSX2, en fase de prototipo.

Se ha seguido el orden alfabético para permitir un fácil acceso, habiéndose marcado las iniciales de las palabras en cada página.

Cuando los términos son palabras clave reconocidas por el ordenador (sea en *BASIC* o en lenguaje ensamblador) se encuentran impresas en mayúsculas. Cuando se trata de términos de uso general que no guardan relación con el sistema, se han escrito en minúsculas (con la excepción de las abreviaturas como *BASIC, ASCII,* etc.).

Dentro del texto o al final de la explicación suelen figurar referencias de las palabrass asociadas, cuya lectura es aconsejable.

Siempre que ha sido posible se han incluido instrucciones en modo directo o pequeñas rutinas para mostrar la acción de cada instrucción, función y operación, cuya ejecución es recomendable, junto con todas las posibles opciones y variantes que el lector pueda imaginar.

Con esta *Microguía* se intenta poner en manos del usuario y programador en MSX, un medio de consulta rápida que le permita resolver sus dudas sobre la programación del equipo y sobre conceptos referentes al mismo que tenga que manejar.



Libro: MSX. 30 rutinas de utilidad en Código Máquina
Autor: Steve Webb
Editorial: Ra-Ma
Páginas: 96

El mundo de MSX necesitaba un libro como éste: sencillo, claro, manejable y eminentemente práctico; destinado a esa inmensa mayoría de lectores (probablemente muy jóvenes) cuya forma principal de disfrutar del uso de su ordenador consiste en realizar sus propios programas (juegos, sin duda) con efectos espectaculares y sin romperse demasiado la cabeza.

Al principio se hace un somero repaso a la estructura de la memoria *RAM* de los MSX, los límites de espacio disponible en ella y el uso de la instrucción *CLEAR*. A continuación nos presenta un programa que nos servirá para introducir el código de las rutinas (digamos, un cargador) y que deberemos conservar en cinta o disco.

Y sin más dilación, comienzan las rutinas. Estas están preparadas de forma que no se solapen unas con otras, lo cual permitirá tenerlas todas juntas en memoria y así poder utilizarlas sin limitaciones en los programas. Por supuesto que no vamos aquí a detallar cada una de ellas, ya que para eso está el libro. Pero para que os hagáis una idea de sus posibilidades, hay rutinas para desplazar el texto en cualquier dirección (izquierda, derecha, arriba y abajo), girar caracteres de texto 90 grados, cambiar el color de fondo de ciertos caracteres, borrar un trozo aislado de la pantalla, caracteres gigantes, pasar una pantalla a memoria y recuperarla de memoria, tablas de puntuación para juegos, efectos sonoros especiales, conseguir que la pantalla desaparezca por el centro, como si hubiera un desagüe, y algunas cosas más.

En resumen, un libro de bolsillo interesante y, como ya hemos dicho, eminentemente práctico.

SOFTWARE

Programa: Chiller
Tipo: Juego
Distribuidor: ABC Soft
Formato: Cassette

Basado parcialmente en un video-clips, nos llega espeluznante, tenebroso y fantástico *Chiller.*

La noche aparecerá llena de un silencio espantoso que te llena de la más implacable soledad. Temes por tí y por tu amada, la cual ha sido raptada por no sé que misterio y se encuentra prisionera en una deshabitada casa de las afueras de la ciudad en un desolado paraje, donde la luna llena hace de ella un espectro viviente.

Existirá una sucesión de pantallas que debes ir pasando con sumo cuidado hasta llegar a la última, en la que se encuentra tu amada, pero los peligros que aparecerán a tu paso, tal vez te hagan desistir

Si llega a tocarte algún energúmeno o cadáver resucitado, espectro o murciélago, te extraerá la energía e irás perdiendo poco a poco fuerza, ya que además tienes que saltar y correr, pero puedes reponerla. A lo largo del juego, verás gran cantidad de setas por todas partes, cómelas y esto te dará más energía, pero no te las comas todas pues algunas son venenosas.

Ya era hora de conocer un juego que tuviese algo de singular en cuanto al tema que trata y a su realización. *Chiller* no es un juegto cualquiera, es una sabia combinación entre el misterio, la aventura, la innovación y diversidad de gráficos.

Este juego figura como uno de los diez más vendidos en el Reino Unido, ya que contiene una serie de características poco comunes entre los demás juegos, y lo más importante es que divierte. Una pequeña innovación que debemos atribuirle es la posibilidad de elegir representante, chico o chica, ya que no siempre debe ser la chica la que sea raptada; vamos, eso le puede suceder a cualquiera, así como la diversidad de elementos que aparecen a lo largo de las pantallas y la determinación con que han sido realizados. En la primera pantalla nos encontramos con un bosque tenebroso, nosotros vamos en nuestro coche de paso, cuando de repente se para y nos obliga a ir en busca de ayuda, nos damos cuenta de que lo importante en ese bosque es sobrevivir y que nuestra amada está en peligro. Para conseguir pasar a la siguiente pantalla tendremos que recoger todas las cruces que se encuentren a tu paso, cruces que te permitirán alejar los monstruos de tí, pero no te olvides de comer setas ya que si no, no podrás moverte y caerás abatido, con lo que el juego habrá terminado. Cada pantalla es diferente y muy variados los temas que se tratan en cada una. Más tarde tendrás que atravesar un «gheto». Hacemos referencia a esta pantalla por ser la más completa en cuanto a realización de gráficos y al color. Los elementos que la componen no son de gran realización ya que concurren en la pantalla gran número de ellos y el agrupamiento no permite una excesiva claridad. Señalaremos también que al transcurrir la aventura durante la noche el fondo negro de la pantalla hace que resalten los más tímidos colores, así se produce un contraste entre los colores predominantes de la noche y el color particular de





los elementos que forman esta aventura.

la cuarta y última, tendrá como escenario la mansión encantada en la que se encuentra encerrada nuestra compañera, debemos rescatarla y defendernos de todos los monstruos que salgan a nuestro paso, pero aquí no termina la aventura: tienes que volver a atravesar las pantallas hasta conseguir llegar a tu coche y marcharos.

Este juego es, por su tema y por su visión de la aventura, diferente, y además proporciona a lo largo de la aventura una serie de alicientes que nos hacen estar pendientes en todo momento de lo que ocurre. Con un poco de habilidad conseguiremos nuestro propósito.

Puntuación:
Presentación: 8
Claridad: 7
Rapidez: 7
Adicción: 8

Programa: Profanation
Tipo: Juego
Distribuidor: Dinamic
Formato: Cassette

Johnny Jones, héroe del *Saimazoon* y del *Babaliba* continúa su reto personal hacia la muerte en esta aventura de singular acción, en la que guiado por el peligro que entraña se sumerge de lleno en los pasadizos de una pirámide de Egipto.

A lo largo de tres mil años, los mejores exploradores han intentado profanar el templo de *Abu Simbel,* construido por *Ramses II* para albergar en él todas las riquezas y secretos que multitud de generaciones han seguido guardando en sus entrañas, sin embargo llegar a la cámara mortuoria es inaccesible y salir con vida del laberinto imposible. Todos los que osaron profanar el Templo jamás regresaron; por ello nuestro héroe debe romper con los tópicos que han acontencido durante tanto tiempo este siniestro pero maravilloso enigma.

Nuestro héroe no será un singular aventurero, que se bata a capa y espada por los entramados laberintos, sino que debido a una maldición de nuestro faraón se ha visto convertido en un perro, esto no quita en absoluto atractivo al juego, ya que al ser más pequeño puede acceder a todos los recovecos ocultos y desafiar a los más temibles animales que aparezcan a nuestro paso. Será un compañero ideal de fatigas, y nos ayudará a tomar una decisión, que a nosotros no se nos haya ocurrido, esta indicación nos la dará con una leve sonrisa o mostrándonos algo.

No queremos que parezca una aventura como las demás, por lo que vamos a comentar los pasos a seguir para que podáis entender cada una de las pantallas de las que consta el juego, muy diferentes todas ellas.



La primera que nos abre la entrada al Templo será una dura prueba de habilidad y valentía, serán numerosos los defensores de la antecámara, desde una cuchilla cortante, hasta gotas sucesivas de ácido que en caso de tocarte te fulminarán, por esto, debes fijar gran atención en la exactitud de los movimientos que realices. Esta advertencia te será dada al comienzo del juego, en un proverbio que dice: «primero calcula y luego actúa».

Numerosos serán todos los ele-

mentos que nos encontremos y cada uno será diferente, variando desde arañas, serpientes, o esferas luminosas que nos desintegrarán en un instante.

No contaremos con tiempo definido para jugar, pero sí tendremos algún impedimento. Contaremos con quince vidas al comienzo y las iremos perdiendo sucesivamente cada vez que seamos atacados por algún guardián de la Pirámide, con la posibilidad de anular todo el recorrido en algunas pantallas, ya que unos nos harán perder una vida, pero otros nos enviarán a nuestro lugar de comienzo.

Mucho tenemos que decir en cuanto a gráficos, ya que su creación corresponde a las técnicas habituales de creación de sprites, detallados y singulares, en esta última característica es donde queremos hacer hincapié. Debido a la originalidad de estos sprites, pueden variar desde escaleras deslizantes a momias que interrumpen el paso, es decir, cualquier cosa que pueda imaginar su mente.

Johnny Jones y Ud. tendrán que descubrir uno por uno todos los enigmas que se encuentran en esta maravillosa aventura.

Puntuación:
Presentación: 7
Claridad: 7
Rapidez: 9
Adicción: 8

Programa: Nick Neaker
Distribuidor: Microbyte
Tipo: Juego
Formato: Cassette

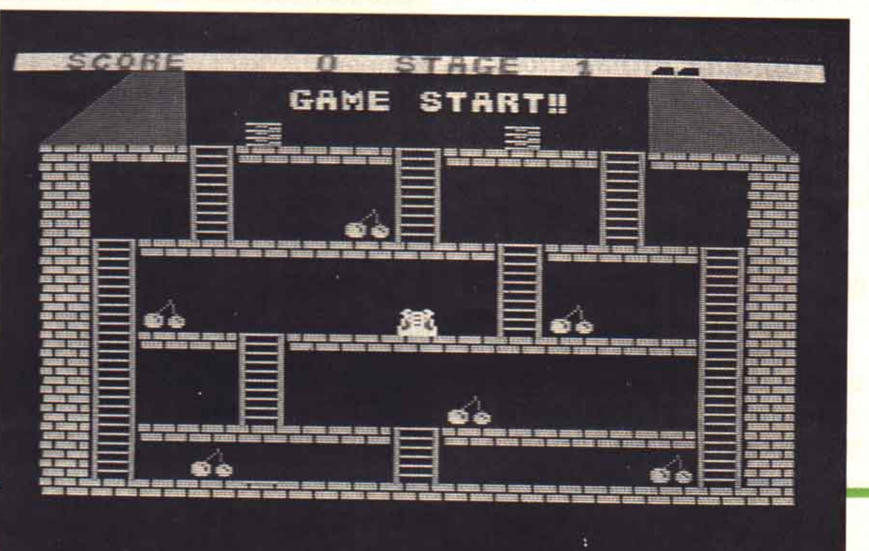
Hay ocasiones en las que un plácido sueño puede convertirse en toda una aventura. Objetos que pasan inadvertidos por tu vida, escapan de tu sueño y vagan por tu mente, para crear un mundo fantástico donde casi nada tiene sentido. Esta noche podría ser, por ejemplo, una zapatilla. Inesperadamente has encontrado en una desconocida casa, habitada por extrañas criaturas sacadas de lo más profundo de tu cerebro y...

Con tu ordenador serán más de un sueño los que descubras por lo que *Nick Neaker*, será una aventura entretenida y extraña, porque ¿cómo una zapatilla puede comerse fresas o aplastar a pequeños seres que te persigues? Imaginación desde luego no falta en este juego.

Te encontrarás en una casa deshabitada en la que tendrás que sobrevivir comiéndote todas las frutas que encuentres a tu paso y aplastando a numerosos seres que te lo impedirán. Conseguir acabar en cada pantalla con todas las frutas que haya no es suficiente para poder pasar a la siguiente, tendrás que pisar también unos muelles que te permitirán correr más deprisa y aplastar a los monstruos, hasta que no pises ningún muelle, tan sólo podrás esquivarlos. Deberás marcarte una táctica. La que te recomendamos es que atraigas la atención de estos monstruitos hasta tí, procurando colocarte cerca de uno de los muelles y así, cuando estén todos en el mismo piso que tú, pisa el muelle y tu zapatilla empezará a saltar como loca en todas las direcciones aplastándolos.

La originalidad de este juego no está en el tema que trata, puesto que podríamos agruparlo en lo que se conoce como «comecocos», sin embargo sí podemos mencionar la capacidad creativa en cuanto a los elementos que utiliza. El principal elemento con el que jugarás será una zapatilla, pero una zapatilla de clase, esperamos que veas su realización en la pantalla. Los movimientos que ésta realiza van acompañados casi siempre de un pequeño sonido, que nos ayudará a saber si ha conseguido una fruta o si ha pisado algún muelle, entonces no podrás controlarla. Ahora bien, mientras que la trayectoria que realices no tenga ninguna «emergencia» podrás dominarlo fácilmente y ascender sin ningún problema a otro piso superior.

En cuanto a los demás elementos que forman el juego, no se trata de lanzarte a comer frutas sin ton ni son, ya que estas se encuentran estratégicamente colocadas y ten-





drás que pensar todos tus movimientos si no quieres que te acorralen.

La realización de los monstruos pensamos que podía haber sido mejor, ya que en comparación con los *sprites* de las frutas y la zapatilla se ven menos detallados, pero comen igual o mejor que muchos otros, así que no os despistéis.

Es de agradecer que en cualquier juego el sonido haga su presencia, y aquí está presente tanto al comienzo del programa como durante su ejecución, y además no resulta desagradable, pues nos acompaña en todos los movimientos que realicemos.

Nick Neaker es un juego de imaginación, pero lo más importante, también es entretenido.

Puntuación:
Presentación: 7
Claridad: 7
Rapidez: 7
Adicción: 7

Programa: Black Jack
Distribuidor: Idealogic
Tipo: Juego
Formato: Cassette

Este juego es uno de los más populares de casinos, pero apartándonos un poco del puro azar, al editar este programa se han tenido en cuenta dos aspectos fundamentales. Al tratarse de un juego que parte de una base de azar, permite poder efectuar múltiples opciones que dependen no solamente de la estrategia a utilizar, sino también del estudio detallado e incluso matemático-estadístico, lo cual permite utilizar el juego como un buen ejercicio de razonamiento. De otra parte, facilitar al apasionado del juego un programa útil para simular partidas en tiempo real, al igual que se dan en cualquier casino de Europa.

El programa reproduce el curso del juego tal y como se realiza en los casinos, pero con algunas diferencias, por ejemplo, no elimina las cinco primeras cartas de cada sabot, permite al jugador barajar cartas antes de cualquier mano; permite pedir carta tras el split de dos ases, etc.

Para una persona que no conozca el juego del *Black Jack* le será un tanto difícil entender estas salvedades, por lo que además de ser un buen juego en su creación, también se le acompaña de un amplio y detallado manual de instrucciones, para aquellos que son principiantes.

Para jugar, tan sólo es necesario conocer las reglas, posturas y modo de apostar. Todo ello nos viene dado al comienzo en un menú, el cual utilizaremos para seguir los pasos del *Black Jack*. Tenemos dos opciones:

—Partida con cartas gráficas, creadas con gran minuciosidad y ayudada de un colorido que nos recuerda las grandes partidas sobre tapete verde.

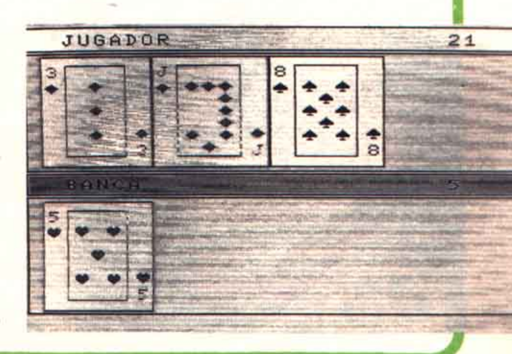
Aquí observamos menos agilidad en el juego, debido al tiempo que tardan en dibujarse en la pantalla, pero de todas maneras es mínimo.

—Partida, en la que sólo tendremos el número de las cartas que obtengamos, así como su valor.

Se podrá determinar el saldo de cada jugador, no superando el millón y realizando apuestas siempre no superiores a 9.999. En cualquier momento puede saber el saldo que le queda a cualquier jugador.

También puede jugar una sola persona contra la Banca. Si actúa correctamente, la ventaja de la Banca es algo inferior al 1%, si no lo hace bien la ventaja de la Banca se agranda enormemente y en menos que esté en convivencia con la fortuna, su ruina se acelerará.

La realización técnica del programa, tanto en gráficos como en reglas, ha sido elaborado por un grupo de técnicos de Dimensión-New, efectuado por un frío cálculo por ordenador y remitiéndose al campo de la Estadística y Matemática para observar todas las posibilidades y facilitar su compren-



el mejor softwa





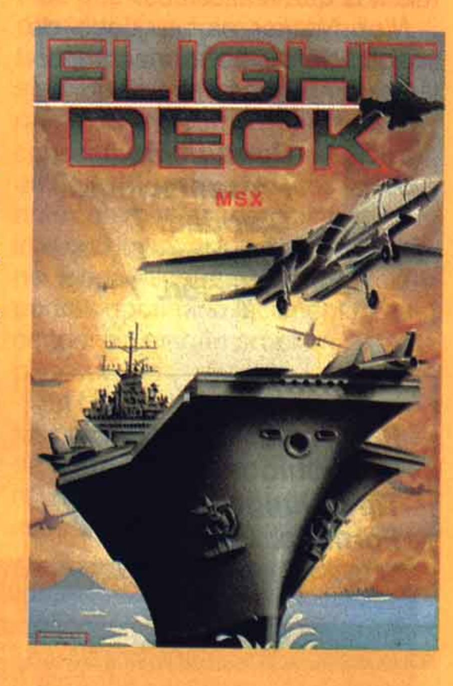
DROME

Entretanto en DROME, un Super-ordenador, debes encontrar y eliminar los sofisticados sistemas de defensa y supervivencia.

Has de elegir uno de los cuatro sectores que constituyen los mecanismos de defensa de esta terrorífica máquina.

Un atractivo juego de acción, donde se pone a prueba la capacidad de la máquina y del jugador.

Precio de venta 2.600 ptas. (IVA incluido)



FLIGHT DECK

Sienta la emoción del golfo de Sidra en casa.

FLIGHT DECK es un juego de estrategia y habilidiad en el que tendrás que desmantelar las bases enemigas.

Al mando de un portaaviones donde dispones de 10 unidades de combate... y poco tiempo.

Precio de venta 2.600 ptas. (IVA incluido)





MC-ATTACK

Ayuda a Fredy, el Rey de la Hamburguesa a preparar el suculento manjar que hace las delicias de los comensales.

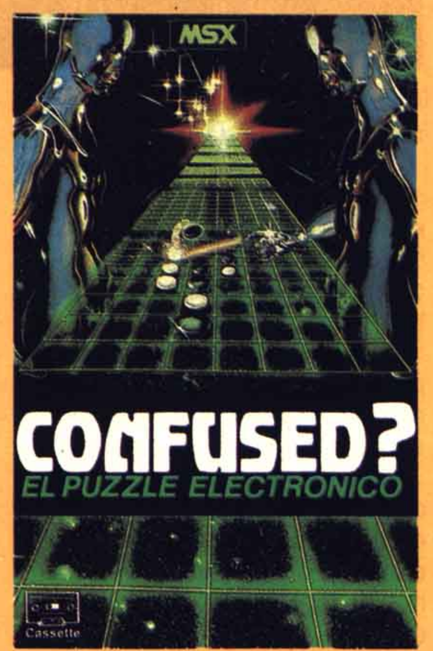
Ten cuidado con las salchichas grasientas y los huevos escurridizos que intentarán arruinar tu exquisito plato.

Defínete con la pimienta y procura hacer el mejor número de hamburguesas posible.

... Buen provecho.

Precio de venta 1.900 ptas. (IVA incluido)

re para MSX



CONFUSED?

Es el puzzle electrónico.

El objeto del juego es resolver 10 puzzles con distinto número de piezas, a elegir, pero todas... MOVIENDOSE.

Pon a prueba tu inteligencia y capacidad de deducción para solucionar algunos de estos entretenidos rompecabezas.

Precio de venta 2.600 ptas. (IVA incluido)

NORTH SEA HELICOPTER

Una explosión en una plataforma en el mar del Norte arroja a los hombres a un destino incierto...

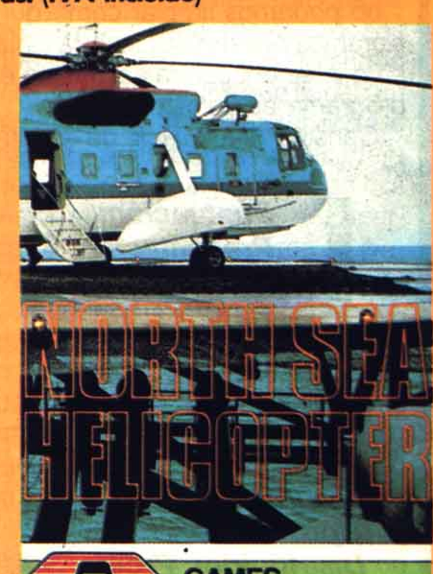
Empieza una carrera contrareloj para salvarles de su fatal situación.

Tienes que convertirte en un piloto experimentado para mantener el control del helicóptero... El tiempo empeora.

¿Crees que cumplirás la misión?

Precio de venta 2.600 ptas. (IVA incluido)









SPACE RESCUE

Desciende de tu nave nodriza a la superficie lunar e intenta rescatar a los hombres extraviados.

Ojo con los meteoritos que te destruirán cuando intentes regresar.

Disponer de nuevas plataformas para culminar con éxito la misión.

Desgraciadamente tu nave nodriza está bloqueada por unidades de combate enemigas... Intenta destruirlas.

Precio de venta 2.600 ptas. (IVA incluido)

Solicita tu programa favorito en tu tienda especializada habitual o recorta y envía este cupón a INFODIS, S. A., c/Bravo Murillo, 377. 5.º A - 28020 MADRID.

NOMBRE ______

DIRECCION ______

POBLACION ______ PROVINCIA ______ C. P. ______

Forma de pago: CHEQUE ☐ CONTRAREEMBOLSO ☐

TARJETA DE CREDITO: American Express ☐ VISA ☐ INTERBANK ☐

Número de mi tarjeta ☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐

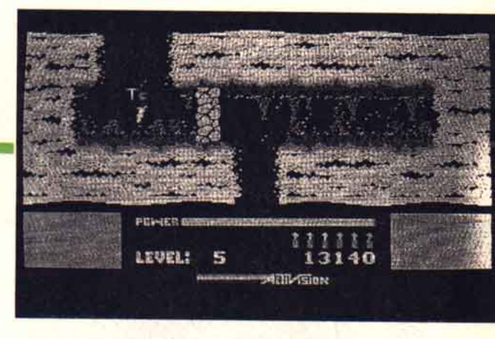
CADUCIDAD ______

Firma ______

TITULO	CANTIDAD	TITULO	CANTIDAD
DROME FLIGH DECK MC-ATTACK		CONFUSED? NORTH SEA HELICOPTER SPACE RESCUE	

infodis, s.a.

SOFTWARE



sión a todos los entusiastas de este juego, pero si la partida es contra su ordenador, es mejor que se apoye en la ciencia que en la suerte.

De todas formas el *Black Jack,* es y será uno de los juegos que atrae más público por las probabilidades que ofrece de poder mejorar la estrategia en cada partida, superando lo que el azar dicte en cada instante.

La presentación del juego por Idealogic se ha hecho de una manera esmerada y se acompaña de una nueva ayuda para aquellos que poseamos numerosos juegos y no podamos limpiar como quisiéramos nuestro cassette, la cinta en que se encuentra así mismo el programa es autolimpiadora.

Consiga *Black Jack,* no es difícil.

Puntuación:
Presentación: 8
Claridad: 8
Rapidez: 7
Adicción: 8

Programa: HERO
Tipo: Juego
Distribuidor: Proein, S.A.
Formato: Cassette

Una experiencia sin igual acaba de comenar y tú eres el protagonista, eres *HERO.*

Un viaje a las profundidades de una caverna en constante tensión, es en líneas generales el tema del juego, pero no es tan sencillo como se nos presenta, la dificultad consiste en que al mismo tiempo que viajamos por ella sin resultar heridos, ayudemos a sobrevivir a aquellos que quedaron encerrados o no soportan la gran prueba de supervivencia.

Equipados de un traje espacial para resistir la falta de oxígeno y el calor, *HERO* debe lograr aquello donde los demás sucumbieron: llegar al fondo de la Tierra. Pero, ¿cómo nos moveremos a lo largo de grutas y cavernas?, nos acompañará un retropropulsor de aire que permitirá nuestro desplazamiento en el espacio de las grutas y nos permitirá no tener que gastar energía a la hora de desplazarnos. De todos modos debemos dosificarla ya que si no nos queda energía cuando encontremos a alguien, no nos será posible ayudarle.

La aventura discurrirá entre la suerte y la intuición. Saber elegir el camino que tomaremos y estar preparados ante cualquier eventualidad deben ser nuestros primeros mandamientos, y sobre todo poder reaccionar a tiempo, ya que el juego comienza con cuatro vidas que se pueden aumentar cada vez que consigamos 20.000 puntos, pero que también se pueden perder en un segundo. Los peligros que nos acechan son infinitos, lo más común es la presencia de muros que nos impedirán el paso, para lo que utilizaremos dinamita. Esto se consigue pulsando el cursor inferior o bien llevando hacia atrás el mando del joystick. Una vez hagamos este movimiento debemos alejarnos lo más rápido y lejos posible, pues sino saltaremos en pedazos nosotros también.

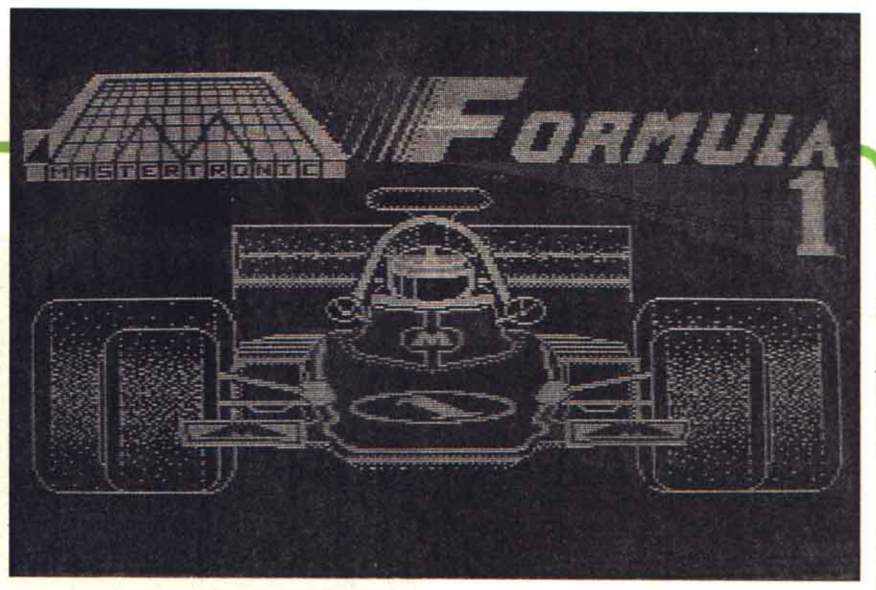
Contaremos con seis cargas explosivas al comenzar el juego, las cuales iremos gastando a lo largo de viaje. Cada vez que consigamos llevar energía a uno de los supervivientes, estas cargas aparecerán de nuevo y comenzará una nueva ruta.

Debemos hacer resaltar la excelente calidad de sonido como elemento importante. La aventura encontrará un momento en que el silencio será sepulcral cuando estemos quietos, pero rápidamente comenzará el ruido de nuestro propulsor, el efecto silbido del rayo láser y la concatenación de sonidos al estallar la dinamita.

El juego es una innovación en cuanto a tema, las características de nuesto HERO, y la creación del contexto. Cada sprite guarda el mínimo detalle y no sólo la diferencia entre ellos se marca por la forma, sino por el color que es totalmente vivo. Podemos mencionar como ejemplo la realización de los muros de lava o ese gran río con el que nos encontraremos, peligros que no hemos enumerado porque cuando comienze esta aventura ya no les importará lo que ocurra hasta que no lleguen al final, y eso es bastante difícil.

Puntuación:
Presentación: 8
Claridad: 8
Rapidez: 7
Adicción: 9

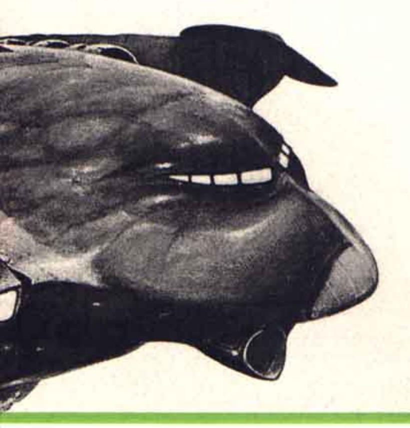




Programa: Fórmula 1 Simulator
Tipo: Juego
Distribuidor: ABC Soft
Formato: Cassette

Este programa-juego simulador de las carreras de Fórmula 1, está basado en los niveles de aceleración, frenado, y agarre al firme de los circuitos de competición en los que se desarrolla la Fórmula 1, lo que nos permite actuar con mayor seguridad a la hora de conducir nuestro prototipo, ya que no tendremos que depender directamente de si las curvas son demasiado cerradas o si los demás corredores se preparan a cerrarnos. Es suficiente con que ejerzamos un control total sobre el acelerador, freno y dirección. Dirigir un bólido de estas características y a estas velocidades no será tarea fácil, ya que debemos tener en cuenta los factores de riesgo que nos acompañan en este juego.

La caja de cambios juega un papel importante, ya que este juego incluye dos posibilidades a elegir: cambio de marcha manual o automático. En el primero, realizaremos una conducción verdadera, ya que tendremos en cuenta las posibilidades del motor como si fuera un coche normal, si no, haremos saltar el motor y nuestra carrera habrá terminado. El segundo, indicado para aquellos que juegan por primera vez, las servirá para reconocer el terreno, los adversarios y la direccionalidad del bólido, pero esto lleva una disminución de la realidad ya que no nos daremos cuenta de los cambios de marcha que se efectúan y cómo nos afectan en el transcurso de la carrera, a no ser que nos guiemos por el sonido que se produce cada vez que se reduce o acelera, elemento a destacar en este juego, ya que una carrera de Fórmula 1 sin el rugir de los motores, se pierde expectación.



Contamos con la posibilidad de elegir entre diez pistas, todas ellas inspiradas en los circuitos más famosos del mundo, con condiciones de ruta mojada o seca. Entre ellos podemos destacar Mónaco, Paul Ricard o Silsverstone.

Conocer un circuito es algo primordial a la hora de conducir por ellos, ya que en el momento que alcancemos mucha velocidad nos será difícil reaccionar ante una curva si no sabemos hacia qué lado crece. Como buen piloto de esta competición, debemos recomendarle que aunque no cuente con un copiloto, realice un planillo con apartados como: circuito y características, revoluciones del motor y velocidad a desarrollar a lo largo del recorrido.

Otra característica importante del juego es la perspectiva: desde nuestro bólido iremos divisando los acontecimientos que surjan en la carrera. En la parte inferior de la pantalla aparecerá el salpicadero de nuestro automóvil y desde esa visión veremos cómo se nos acercan los contrarios o cómo se cierran las curvas. Tengamos mucho cuidado en lo referente a esto, pues las curvas no terminan donde nosotros creemos. Mantener una distancia equilibrada a las rayas de los laterales es imprescindible para lograr no salirnos de la pista.

Disfrute de los paisajes, pero no espere encontrar pantallas de gran resolución, en las que el mínimo detalle se diferencia.

Tome en cuenta los puntos de frenado y la línea a seguir en las curvas, mantenga el coche en el límite de adherencia de los neumáticos, adelantando si puede y experimente al máximo la velocidad y excitación de una carrera de coches.



Puntuación:
Presentación: 6
Claridad: 7
Rapidez: 7
Adicción: 7

Interiorid

de los SVI-328

Empezamos a ver en este capítulo la forma de usar el VDP y la VRAM en cada uno de los tres modos de pantalla.

Si habéis hecho el experimento que propuse en la capítulo del mes pasado habréis obtenido un resultado parecido al de la figura 3.

Comparando la tabla de la figura 3 con la tabla de la figura 2 que vimos el mes pasado, podemos ir descubriendo cómo está organizada la *VRAM* en los distintos modos de pantalla de nuestro ordenador.

SCREEN Ø. El modo de texto

Viendo el contenido de los registros 0 y 1 en la figura 3 podemos ver que estamos en el modo *SCREEN 0* con una *VRAM* de 16 *Kbytes* con la pantalla y las interrupciones activadas y con el desconocido vídeo externo desactivado.

En el registro 2 tenemos un 0



ades



y SVI-318

que nos indica que la tabla de denominaciones comienza en la dirección 0 de la *VRAM.*

Para hablar de la tabla de denominaciones vamos a ver antes cómo está organizada la pantalla en el modo 0 de nuestro ordenador.

La pantalla en el modo 0

En este modo la pantalla está dividida en 24 filas de 40 columnas y cada una de estas 24*40 posiciones contiene un carácter de 8*6 *pixels.* La figura 4 muestra dicha distribución.

La tabla de denominaciones

La tabla de denominaciones tiene 960 *bytes,* (numerados de 0 a 955), y hay una corespondencia directa entre cada *byte* de la tabla y cada una de las casillas de la pantalla mostradas en la figura 4, esta correspondencia consiste en que las casillas de la pantalla representan el carácter cuyo código se guarda en el octeto correspondiente de la tabla de denominaciones, así tenemos que el octeto 0 de la tabla contendrá el código del carácter que aparecerá en la casilla 0 de la pantalla, el octeto 1 el de la casilla 1 y así hasta el 959.

Por esto podemos decir que los octetos de la tabla «denominan» o dan «nombre» a cada una de las casillas de la pantalla.

El código que contienen los octetos de la tabla no tiene una relación directa con el código *ASCII* como podría suceder, sino que se refiere a la posición que ocupa en la tabla de generación de patrones el patrón del carácter representado.

La tabla de generación de patrones

El registro 4 contiene un 1 que nos indica que la tabla de generación de patrones comienza en la dirección 800 hex de la *VRAM.*

Al encender el ordenador o al volver al modo de texto mediante el *BASIC,* el sistema operativo *BASIC* coloca en la tabla de patrones el juego inicial de caracteres.

Cada carácter viene definido por un bloque de ocho octetos en la *VRAM.*

El número de máximo de caracteres que se pueden definir en la tabla es de 256. Esto es así porque en los octetos de la tabla de denominaciones solamente se podrá guardar un número entre 0 y 255.

Sin embargo hay un medio de tener más caracteres definidos en la *VRAM,* es más, podemos tener hasta siete juegos de 256 caracteres cada uno definidos en la *VRAM* y el acceso a los mismos es tan fácil como colocar un número entre 1 y 7 en el registro 4 del *VDP.*

Hemos visto que el comienzo de la tabla del generador de patrones se encuentra multiplicando el contenido del registro 4 por el número 800 hex. Si variamos entre 1 y 7 el contenido de este registro tendremos 7 diferentes direccio-

nes donde puede comenzar la tabla.

La longitud de la tabla es fácil de hallar, 256 caracteres de 8 octetos ocupan 2048 *bytes* = 800 hex *bytes,* como anillo al dedo, pues el registro cuatro se multiplica precisamente por 800 hex.

El mapa de la VRAM en el modo 0

La figura 5 nos muestra un mapa de la VRAM en el modo de texto donde podemos ver los siete juegos de caracteres.

El juego inicial de caracteres

Al encender el ordenador hemos visto que el juego de caracteres usado es el que comienza en la dirección 800 hex y es el único que está definido, pues los otros seis han de ser definidos por el usuario.

Para ver los caracteres que componen este juego teclea la siguiente línea de programa:

```
FOR A=0 TO 225:VPOKEA,A:
NEXT
```

tras pulsar *ENTER* verás aparecer en la pantalla los 256 caracteres que están definidos en un principio en el ordenador si no los has cambiado tú ya.

El ordenamiento seguido para la definición de este juego de caracteres inicial es el siguiente.

– De 0 a 94 se definen los 95 caracteres cuyos códigos *ASCII* están comprendidos entre 32 (que representa un espacio) y 126 (el símbolo ˜).

– El 95 es el código del cursor inverso.

– De 96 a 190 están los caracteres inversos que equivalen a los



códigos *ASCII* entre 32 y 126 sumados a 64.

– El código 191 es el cursor normal. Si después de haber introducido la línea *BASIC* antes mencionada, con las teclas de movimiento del cursor desplazas éste por encima de los caracteres que han aparecido, verás que el carácter de la posición 191 refleja también el carácter por encima del cual está pasando el cursor.

– Las casillas 192 a 247 las ocupan los 56 caracteres gráficos cuyos códigos *SVI* están comprendidos entre 160 y 215.

Cómo definir los otros juegos

Para definir los otros juegos tendríamos que proceder de la siguiente manera.

- Elegir una de las siete posibles zonas para definir nuestro juego de caracteres. Ej.: la zona 2.
- Multiplicar el número de zona escogido por 800 hex. O sea 2 * 800 hex. = 1000 hex.
- En las direcciones 1000 hex a 1007 hex de la *VRAM* meteríamos los 8 números que definirían el primer carácter de nuestro juego.
- En las direcciones 1008 hex y 100f hex definiríamos el segundo carácter. De 1010 hex a 1017 hex el tercero y así sucesivamente hasta completar todos los caracteres.

Para hacer todo esto más fácilmente, podéis introducir en vues-

FIGURA 3

LOS REGISTROS DEL VDP EN LOS TRES MODOS DE PANTALLA

Registro	0	1	2	3	4	5	6	7	8
MODO 0	0	F0hex	0	NO U.	1	NO U.	NO U.	F4hex	?????
MODO 1	2	E0hex	6	80hex	0	36hex	7	?????	?????
MODO 2	0	E8hex	2	NO U.	0	36hex	7	?????	?????

tro ordenador el programa que viene a continuación.

El programa está pensado para trabajar en un *SV-328* que además del cassette tiene conectada una unidad de disco. Si no tenéis unidad de disco conectada y sobre todo si lo queréis ejecutar en un *SV-318* tenéis que cambiar lo siguiente:

- En las líneas 10, 440, 450, 580 y 590 cambiad &hCDB0 por &hED00 y en la línea 590 cambiad &hD5B0 por &hF500.

Explicación del programa

El programa está dividido en cinco partes:

- La primera va de la línea 10 a la 30 y reserva un espacio de memoria, que usaremos después para cargar y grabar datos, además de gestionar las insterrupciones de *«STOP»* y de las teclas de función.
- La segunda abarca las líneas 40 a 130 y es la que presenta la pantalla inicial o de menú.
- La tercera comprende las líneas 140 a 350 y es el núcleo del programa que permite la introducción y cambio de los caracteres directamente en su zona de ubicación.
- La cuarta parte abarca las líneas 360 a 470 y se ocupa de la carga de caracteres grabados anteriormente con el programa en disco o cinta.
- Y la quinta abarca las líneas 480 a 640 y se encarga de la grabación de caracteres en cinta o disco.

Manejo

Copia el programa en tu ordenador teniendo en cuenta los, cambios antes mencionados si es un *SV-318*, y antes de hacer cual-

```
FIGURA 4

LA PANTALLA EN EL MODO 0
------------------------

-------------------------------        ------------------------------------
| 0 | 1 | 2 | 3 | 4 | 5 |<             < 33| 34| 35| 36| 37| 38| 39|
|---|---|---|---|---|---|->             >--|---|---|---|---|---|---|
| 40| 41| 42| 43| 44| 45|<             < 73| 74| 75| 76| 77| 78| 79|
|---|---|---|---|---|---|->             >--|---|---|---|---|---|---|
|/\/\/\/\/\/\/\/\/\/\/\/\/\/             </\/\/\/\/\/\/\/\/\/\/\/\/\/\/

|\/\/\/\/\/\/\/\/\/\/\/\/\/\             /\/\/\/\/\/\/\/\/\/\/\/\/\/|
|880|881|882|883|884|885|8>            >13|914|915|916|917|918|919|
|---|---|---|---|---|---|<             <---|---|---|---|---|---|---|
|920|921|922|923|924|925|92>           >53|954|955|956|957|958|959|
```

quier otra cosa grábalo por si los duendes.

Si todo va bien, al hacer *«RUN»* aparecerá el menú donde se podrá elegir entre cuatro opciones, la única que puede presentar dificul-



tad de entendimiento en un principio es la opción 2.

Al escoger la opción 2 del menú nos introducimos en la parte del programa que nos sirve para definir o alterar los caracteres de los siete juegos de que disponemos.

En un principio la pantalla nos mostrará el juego de caracteres número 1. Para cambiar de juego usaremos los *PFKs* o teclas de función programables, como muestran las descripciones de las mismas en la pantalla, la tecla de función número 8 nos sirve para retornar al menú.

Mediante las teclas de cursor podemos mover el mismo a través del juego de caracteres. Los caracteres por los que va pasando el cursor aparecen ampliados y representados mediante ceros y unos en un recuadro a la derecha de la pantalla, si deseamos modificar el carácter pulsamos *«ENTER»* con lo que el cursor se nos trasla-

FIGURA 5

DISTRIBUCION DE LA VRAM EN EL MODO 0	
DIR. HEX.	CONTENIDO
0000 03BF	TABLA DE DENOMINACIONES 960 bytes. Cada byte se corresponde con una posición de la pantalla y guarda el código del caracter presentado en ella.
03C0-07FF	NO USADO
0800 0FFF	TABLA DEL GENERADOR DE PATRONES Juego de caracteres inicial o número 1
1000-17FF	Juego de caracteres número 2
1800-1FFF	Juego de caracteres número 3
2000-27FF	Juego de caracteres número 4
2800-2FFF	Juego de caracteres número 5
3000-37FF	Juego de caracteres número 6
3800-3FFF	Juego de caracteres número 7

da ahora al recuadro pequeño, otra vez con las teclas de cursor lo movemos a través del recuadro y cuando esté encima del punto que queremos cambiar pulsamos la barra espaciadora con lo que dicho punto pasará a ser uno si era cero y viceversa.

El número debajo del recuadro pequeño nos indica la posición del carácter dentro del juego.

Si queremos pasar enteramente un juego de caracteres a otra zona, lo haremos mediante una grabación previa del juego en cinta o disco y la posterior carga en la nueva zona, mediante las opciones 1 y 3 del menú.

SCREEN1. Gráficos de alta resolución

Mirando las figuras 3 y 2, podemos ver en las tablas el contenido



de los registros en el modo 1.

En los dos primeros registros se han cambiado respecto al modo 0 el *bit* 1 del registro 0 que pasa a contener un uno y el *bit* 4 del registro 1 que ahora contendrá un cero.

Por el contenido del registro 2 tenemos situada la tabla de denominaciones en la dirección 1800 hex de la *VRAM*.

La tabla de denominaciones tiene aquí las mismas funciones que en el modo de texto, pero su aplicación es algo diferente. Para verlo mejor vamos a empezar, como en el modo 0, por ver cómo está formada la pantalla.

La pantalla en el modo 1

Ahora la pantalla está más orientada a manejar cada *pixel* por separado que incluido en un carácter.

Está dividida en 24 filas y 32 columnas por lo que los 256 * 192 *pixels* de la misma están agrupados en bloques de 8*8 *pixels*.

A su vez las 24 filas están divididas en tres secciones de 8 filas cada una. A cada una de estas secciones la llamaremos tercio alto, tercio medio y tercio bajo de la pantalla.

```
DEFINIDOR DE LOS JUEGOS DE CARACTERES

10 CLEAR100,&HCDB0:SCREEN0,1:LOCATE,,0:DEFINTA-J:
ONSTOPGOSUB630:STOPON
20 KEY1,"Juego 1":KEY2,"Juego 2":KEY3,"Juego 3":KEY4,
"Juego 4":KEY5,"Juego 5":KEY6,"Juego 6":KEY7,"Juego 7":
KEY8,"menu":KEY9," ":KEY10," "
30 ONKEYGOSUB270,280,290,300,310,320,330,350:KEYON
40 CLS:LOCATE7,3:PRINT"DEFINIDOR DE CARACTERES"
50 LOCATE7,4:PRINT"========= == =========="
60 LOCATE3,7:PRINT"1.- CARGAR CARACTERES GRABADOS"
70 LOCATE3,9:PRINT"2.- DEFINIR NUEVOS CARACTERES"
80 LOCATE3,11:PRINT"3.- SALVAR CARACTERES"
90 LOCATE3,13:PRINT"4.- SALIR DEL PROGRAMA"
100 LOCATE3,17:PRINT" Elije una opcion"
110 K$=INKEY$:IFK$=""THEN110
120 IFASC(K$)<49ORASC(K$)>52THEN110
130 ONVAL(K$)GOTO360,140,480,630
140 CLS:FORA=0TO15:FORB=0TO15
150 VPOKE40*(A+2)+B+5,16*A+B:NEXT:NEXT:A=0:B=0:C=0:D=0:G=1
155 LOCATE4,19:PRINT"'ENTER'...Selecciona caracter":
PRINTTAB(4)"'SPACE'...Cambia caracter":PRINTTAB(4)"'PFK S'
...Seleccionan juego"
160 J=STICK(0):A=A-(J>0)*(J<3)+(J>7)+(J>3)*(J<7):
B=B+(J>1)*(J<5)+(J>5)
170 A=A*(A>0)*(A<16)-15*(A>15):B=B*(B>0)*(B<16)-15*(B>15)
180 PS=40*(A+2)+B+5:CR=16*A+B:VPOKEPS,&HBF:GOSUB210:
VPOKEPS,CR
190 IFINKEY$=CHR$(13)THENGOSUB220
200 LOCATE25,16:PRINTSTRING$(4-LEN(STR$(CR)),32)+STR$(CR):
GOTO160
210 FORE=0TO7:Z$=BIN$(VPEEK((CR)*8+G*&H800+E)):LOCATE24,E+4:
PRINTLEFT$(STRING$(8-LEN(Z$),48)+Z$,6):NEXT:RETURN
```

```
220 VPOKEPS,CR:I=STICK(0):C=C-(I>0)*(I<3)+(I>7)+(I>3)*(I<7):
D=D+(I>1)*(I<5)+(I>5)
230 C=C*(C>0)*(C<8)-7*(C>7):D=D*(D>0)*(D<6)-5*(D>5)
240 PB=24+40*(C+4)+D:VPOKEPB,&HBF:GOSUB210
250 Y$=INKEY$:IFY$=" "THENVPOKE(CR)*8+G*&H800+C,
VPEEK((CR)*8+G*&H800+C)XOR2^(7-D)
260 IFY$<>CHR$(13)THEN220ELSERETURN
270 G=1:GOTO340
280 G=2:GOTO340
290 G=3:GOTO340
300 G=4:GOTO340
310 G=5:GOTO340
320 G=6:GOTO340
330 G=7
340 Z=INP(&H85):OUT&H81,G:OUT&H81,(4OR&H80):RETURN
350 RETURN40
360 CLS:LOCATE7,3:PRINT"CARGA DE CARACTERES"
370 LOCATE7,4:PRINT"===== == =========="
380 LOCATE3,6:INPUT"De cinta o de disco (c/d) ";L$:PRINT
390 IFL$="d"ORL$="D"THENM$="1:"ELSEM$=""
400 LOCATE3,8:INPUT"En que zona lo cargo (1-7) ";N$
405 IFN$=""THEN400
410 IFASC(N$)<49ORASC(N$)>55THEN400
420 HI=VAL(N$)*&H800
430 LOCATE3,10:INPUT"Nombre del juego ";O$:
IFO$=""ORO$=" "THEN430
440 M$=M$+O$:BLOADM$,&HCDB0:LOCATE3,12:
PRINT"Aguarda un momento"
450 FORVR=0TO&H7FF:VPOKEHI+VR,PEEK(&HCDB0+VR)
470 NEXT:LOCATE3,15:PRINT"Carga terminada :GOTO610
480 CLS:LOCATE7,3:PRINT"GRABACION DE CARACTERES"
490 LOCATE7,4:PRINT"========= == =========="
500 LOCATE3,6:INPUT"Cinta o disco (c/d) ";L$:PRINT
510 IF L$="d"ORL$="D"THENM$="1:"ELSEM$=""
520 LOCATE3,8:INPUT"Numero de juego a grabar (1-7) ";N$
525 IFN$=""THEN520
530 IFASC(N$)<49ORASC(N$)>55THEN520
540 HI=VAL(N$)*&H800
550 LOCATE3,10:INPUT"Que nombre le pones ";O$:
IFO$=""ORO$=" "THEN550
560 LOCATE3,12:PRINT"Aguarda un momento"
570 M$=M$+O$:FORVR=0TO&H7FF
580 POKE&hCDB0+VR,VPEEK(HI+VR)
590 NEXT:BSAVEM$,&HCDB0,&HD5B0
600 LOCATE3,15:PRINT"Grabacion terminada"
610 PRINT:PRINT"Pulsa una tecla para volver al menu"
620 IFINKEY$=""THEN620ELSE40
630 DEFUSR0=&h3498:Z=USR0(0):SCREEN0,1:LOCATE,,1
640 END
```


GANE
7.000 PTAS.
todos los meses
PARTICIPANDO
EN NUESTRO
CONCURSO
MSX Magazine premiará cada mes los programas que nos hagan llegar nuestros lectores.
Para participar en este concurso abierto, todo aficionado a los ordenadores con este estándar deberá hacer llegar a la redacción de la revista el listado, un cassette y un texto explicativo.
Entre todos los programas que recibamos cada mes, serán seleccionados para su publicación aquellos que reúnan los siguientes criterios:
— Originalidad de la aplicación.
— Simplicidad del método de programación.
La única condición para participar en el concurso será que los programas no hayan sido publicados previamente en ninguna revista.
Enviar vuestros programas a: MSX Magazine
C/Bravo Murillo, 377 - 5.° A 28020 MADRID
BANCO DE ESPAÑA
2E1554046
MIL pesetas
CINCO MIL pesetas
14457150





La interconexión del mundo analógico y digital, juega un papel muy importante en la evolución de la informática. Los ordenadores del futuro serán aquellos capaces de entender nuestro habla, estudiar cualquier fenómeno que nos rodea, etc. Estos fenómenos naturales, periódicos o no, son función del tiempo y toman diversos valores. Los ordenadores internamente sólo reconocen el 1 y el 0 como diferentes estados, y no el 5, el 6 ó cualquier otro valor. La única forma de que el ordenador entienda estos estados es por medio de un convertidor analógico/digital *(A/D)*. Los convertidores, *A/D* tienen cientos de aplicaciones, y algunas de ellas

Osciloscopio-Anali

son las de transformar a nuestro ordenador en un osciloscopio, un voltímetro, un termómetro,... No sólo los famosos ordenadores, como los PC, tienen este tipo de *hardware*. El *NEOS VC-10* es un claro ejemplo. Creado para MSX, es capaz de trabajar como un osciloscopio de doble trazo o un analizador de espectros. Viene presentado en un cartucho —más alto de lo normal— y su manejo es muy sencillo: basta con el uso del teclado del cursor y la barra espaciadora (o en su lugar un *joystick* o un ratón) y la tecla *return*.

Menú de opciones y presentación

Tras encender el ordenador, observamos que la pantalla se transforma en una especie de instrumento de laboratorio.

Abajo, quedan una serie de recuadros que componen el menú de opciones. A la derecha, están los controles del aparato, a los que se accede por el menú anterior, y a modo de potenciómetros deslizantes nos indican los valores de los parámetros que podemos variar. El resto de la pantalla está ocupada por una gran ventana a través de la cual se representarán las diferentes operaciones del sistema.

Para elegir una opción solamente hay que trabajar con las teclas del cursor, hasta que se «ilumine» el recuadro deseado. Una vez hecho esto, hay que pulsar la barra espaciadora para modificar los valores que afectan a este recuadro, apareciendo un menú específico, dentro del cual el parámetro se cambia con el cursor arriba/abajo, y el valor con las teclas izquierda/derecha. Para volver al menú principal hay que pulsar de nuevo la barra espaciadora.

Las diferentes opciones son:

1) **Ajuste de la base de tiempos:** Modifica la base de tiempos entre 2 y 49.2 ms (mili-segundos). De esta forma podemos elegir el número de períodos a visualizar en función de la frecuencia de la señal de entrada.

2) **Nivel de entrada:** Controla la sensibilidad de los canales izquierdo y derecho. Puede variar entre 1 (mayor sensibilidad) y 4 (menor sensibilidad).

3) **Tiempo de mantenimiento:** Varía desde 0 a 1483 ms en pequeños saltos. Su misión es mantener el oscilograma realizado durante el tiempo escogido. Tras dicho intervalo, se representará otro oscilograma, para mantenerlo de nuevo, y así sucesivamente.

4) **Ajuste automático:** Realiza un ajuste automático de los valores de la base de tiempos y sensibilidad del aparato, según las señales inyectadas en éste.

5) **Desplazamientos de las curvas del oscilograma:** Sirve para ajustar las posiciones **x** e **y** de las curvas representadas en



dio) y un generador de ruidos. Es posible modular en amplitud cada una de las tres señales mediante un diente de sierra, un diente de sierra invertido o una onda triangular —cuya frecuencia es la mitad que la de las anteriores—. El período de estas ondas trabaja entre 0.14 y 9362 ms.

9) **Elección de la fuente:** Sirve para decidir cual va a ser la fuente de la que vamos a realizar el oscilograma. Puede ser de un micrófono (de condensador incorporado) o unas entradas de línea (independientes). Lo más destacable es que las entradas pueden ser diferentes en cada canal, es decir, podemos aplicar la señal del micrófono al canal derecho y la de línea al canal izquierdo, obteniendo los oscilogramas simultánea-

na básicamente un cartucho que realice estas funciones. Veamos el osciloscopio. En primer lugar distinguimos en el diagrama de bloques un amplificador. Este se en-



zador NEOS VC-10

cada canal. La posición **y** se puede modificar independientemente o a la vez, mientras que la posición **x** (llamada **TIME**) desplaza ambas ondas a la vez.

6) **Color:** Permite modificar el color del borde, de la onda y de la ventana entre los 16 posibles del MSX.

7) **Disparo:** Este ajuste es útil para mantener la estabilidad horizontal del oscilograma. Varía su nivel entre 0 y 11.

8) **Oscilador:** Esta opción está directamente vinculada al generador de sonidos del MSX. Se pueden controlar dos tonos (la frecuencia de estos puede variar entre 0.026 y 111.86 *KHz,* cubriendo sobradamente el espectro de au-

mente. Así, el *VC-10* se convierte en un auténtico osciloscopio de doble trazo.

10) **Modo de funcionamiento:** Con esta casilla podremos elegir cómo queremos que trabaje el sistema. Se distinguen el modo osciloscopio y el modo analizador de espectros. Dentro del primero, es posible utilizar un trazo punto a punto o en forma de línea continua, el trazo puede ser a su vez rápido o lento. El segundo modo, analizador de espectros, puede trabajar rápida o lentamente también (luego veremos en qué consiste esto).

El interior

Vamos a explicar cómo funcio-



carga de elevar el nivel de la señal y adaptarla para que pueda ser introducida en un segundo paso,

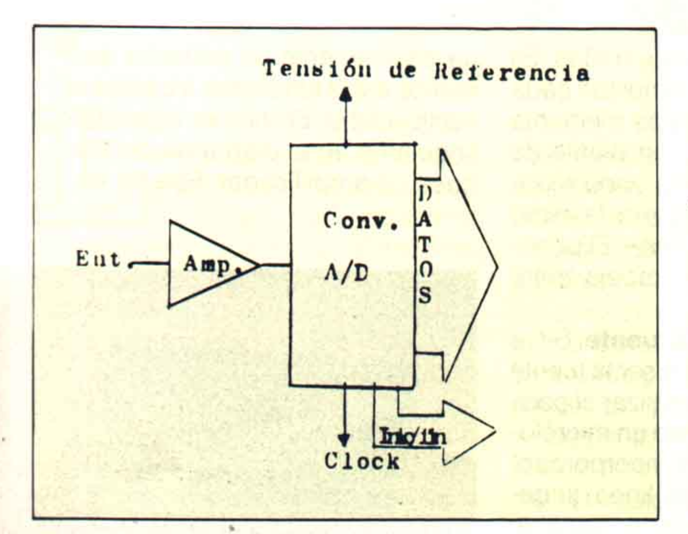


Diagrama de bloques del osciloscopio.

Doble trazo con uno y dos convertidores.

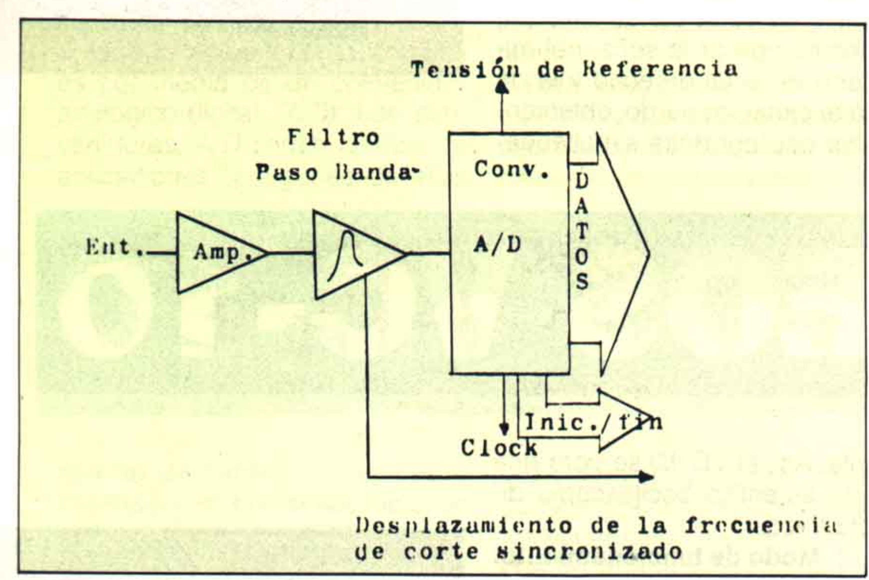


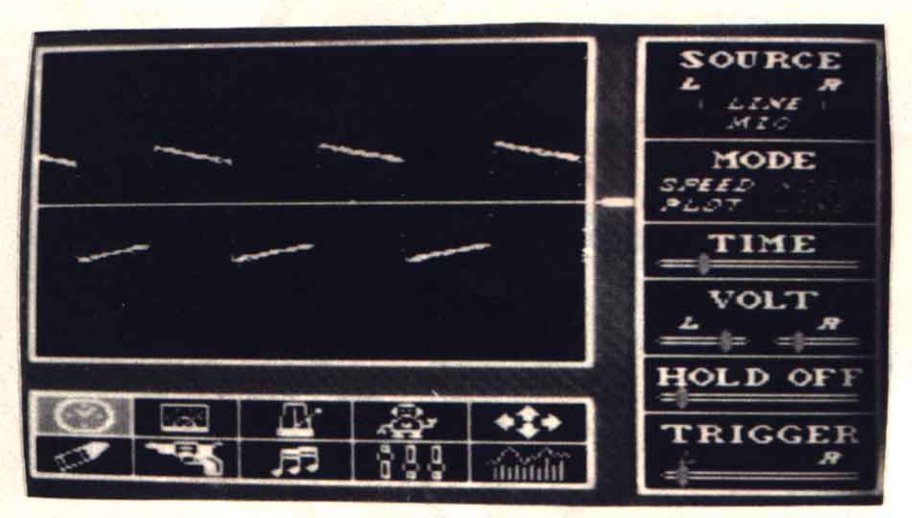
Diagrama de bloques del analizador de espectros.

que es el convertidor *A/D*. Este recibe la señal continuamente, y cuando el ordenador está preparado, le manda una señal que le indica que inicie la conversión. Una vez realizada ésta, el convertidor avisa al ordenador que toma los valores resultantes de la conversión a través del bus de datos. El resultado será un número en binario que será almacenado en una serie de posiciones de memoria, que el ordendor procesará convenientemente. En este caso, en función de dicho valor será dibujado un punto sobre la pantalla.

En el osciloscopio de doble trazo el funcionamiento sería el mismo. Cabría la posibilidad de utilizar un sólo paso *A/D*, y tomar muestras de las dos señales de entrada alternativamente; o bien, utilizar dos pasos *A/D* y que fuera el ordendor quien tomara, también de forma alternativa, los valores de los buses de datos de los dos convertidores.

Estudiemos ahora el analizador de espectros. Vemos que el diagrama de bloques es prácticamente igual, pero entre el amplificador y el convertidor hay una etapa nueva. Se trata de un filtro de paso de banda, se encarga de que sólo pase un estrecho grupo de frecuencias. El filtro tomará en principio su frecuencia de corte (frecuencia que en teoría es la única que deja pasar) en el extremo audible más bajo y el convertidor *A/D* pasará al ordenador el nivel de esta frecuencia. Este, y de acuerdo con dicho valor, representará una banda de mayor o menor altura en la pantalla. El ordenador indicará ahora el filtro que suba su frecuencia de corte, el convertidor volverá a realizar el análisis de la amplitud de esta nueva banda. Este proceso se repite hasta que el filtro toma la frecuencia de corte más alta, y de nuevo se vuelve a empezar.

La diferencia entre un análisis de espectro lento y rápido, consiste en que el primero cubre el espectro tomando más valores que el segundo. De esta forma el análisis rápido valorará menos fre-



cuencias (se puede incrementar el ancho de banda del filtro, es decir, que deje pasar más frecuencias y que analice el nivel de todas ellas).

Todas éstas operaciones corresponden más o menos con el interior del aparato, donde se distinguen convertidores, amplificadores operacionales, y la lógica adicional para trabajar con el ordenador. Una memoria *EPROM* de 16 *Kbytes* tiene en su interior todo el «sistema operativo» del producto, que controla las diferentes opciones.

Conclusión

Se echan de menos algunos detalles que harían del *VC-10* algo más profesional, como serían las diferentes escalas y graducaciones de la pantalla y los controles. Sin embargo, sin ser un aparato de laboratorio, es una de las aplicaciones en *hardware* más interesantes vistas hasta ahora para el MSX.

José Luis Crespo Dueñas

NOTA: El periférico que hemos tratado en esta sección, aún no se comercializa en España. Sin embargo, cabe la posibilidad de que *SERMA,* lo traiga en un futruo no muy lejano.



Pero supongamos que queremos hacer algo tan sofisticado como un listín telefónicoo de todos nuestros amigos, que contenga el nombre, la dirección y el número de teléfono. Es evidente que nos interesa disponer de tres variables:

Variable 1: Nombre
Variable 2: Dirección
Variable 3: Teléfono

y que estas variables deben contener: la primera los nombres de todos nuestros amigos, la segunda ıas direcciones de todos ellos y la tercera sus números de teléfono.

Podremos hacer esto utilizando VARIABLES CON SUBINDICE. Una variable con subíndice es algo así como una variable múltiple, es decir, una variable que comprende otras muchas variables. Imaginemos que tenemos un archivador con tres cajones. En cada cajón vamos a tener tarjetas de forma que la primera tarjeta del primer cajón contenga el nombre del amigo n.° 1, la primera tarjeta del cajón número dos contenga la dirección del mismo amigo y la

Variables con subíndices

Hemos visto ya que en BASIC disponemos de 4 tipos de variables: Enteras, reales de simple y doble precisión y cadenas de caracteres. Esto nos permite realizar programas manejando muchas variables.

Listín telefónico

primera tarjeta del tercer cajón su número de teléfono. La segunda tarjeta de cada cajón contendrá los datos correspondientes al amigo n.° 2 y así sucesivamente. Pues bien, cada cajón se corresponde con una variable con subíndice y cada una de las tarjetas es una variable contenida en la variable con subíndice correspondiente.

Las variables son subíndices, pueden ser también de tipo entero, real de simple o doble precisión o de cadenas de caracteres o alfanuméricas.

Para identificar cada una de las variables contenidas en una variable con subíndice, le asignaremos un número entero, que escribiremos a continuación del nombre de la variable encerrado entre paréntesis. Así, por ejemplo, si llamamos *NB$* a la variable que contendrá los nombres de nuestros amigos, el nombre del primero estará contenido en *NB$*(1), el del segundo en *NB$*(2), y así sucesivamente.

El ordenador necesita que

E

"ECUACION DE SEGUNDO GRADO: AX^2+ BX + C"

A, B,C

D= B * B - 4 * A *C

D<O

SI

NO

D = SQR (D)
X1 = (-B+D)/2/A
X2= (-B-D)/2/A

X 1
X 2

¿DESEAS CONTINUAR?

A $

A$=SI

SI

NO

"RAICES IMAGINARIAS"

FIN

FIGURA 6a

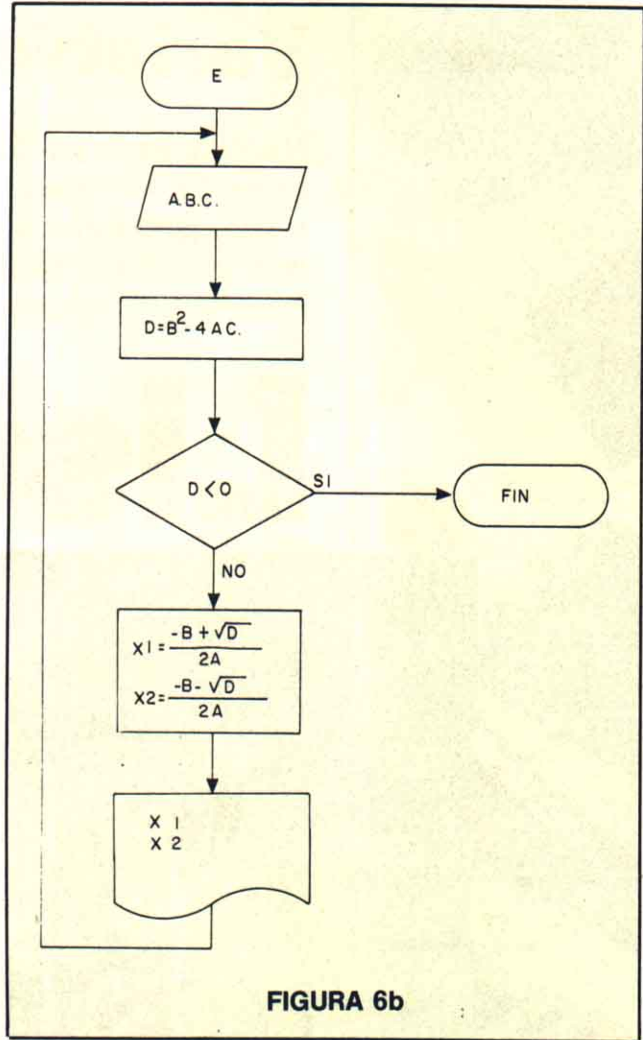


FIGURA 6b

hasta ahora tenemos de *BASIC,* podemos resolver este problema realizando el programa de la figura 7(b), cuyo diagrama de flujo se representa en la fig. 7 (a).

BASIC, sin embargo, posee una potente instrucción con la que se pueden suprimir las líneas 20, 130, 140 y 150. Esta es la instrucción *FOR... NEXT* o *FOR... TO... STEP... NEXT.*

Esta instrucción tiene la siguiente forma.

```
10 FOR I = A1 TO A1 STEP P
.
.
.
```

SERVICIO DE EJEMPLARES ATRASADOS

ESTOS SON LOS EJEMPLARES DE MSX MAGAZINE APARECIDOS EN EL MERCADO CON UN RESUMEN DE SU CONTENIDO



Núm. 1
¿Qué es el MSX? Su BASIC, periféricos, programas, software.



Núm. 4
Las comunicaciones entre ordenadores, la jerga informática, trucos, rincón del lector.



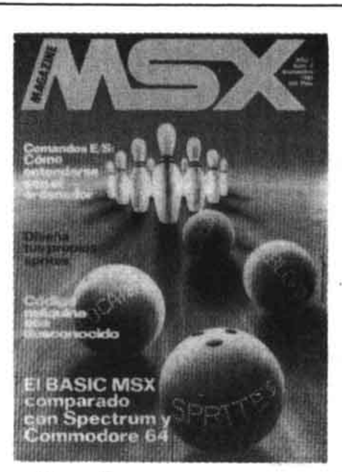
Núm. 6
Los 8 magníficos (test gigante), el bus de expansión, los misterios de la grabación, programas.



Núm. 8
Compact Disc, el periférico del futuro. Test: Dynadata DPC-200. Continuamos con la memoria de video. Libros, software, programas, trucos.



Núm. 2
Generación de sonido, MSX-DOS, el ordenador por dentro, programas, noticias.



Núm. 5
Comandos de entrada/salida, el BASIC MSX comparado con Spectrum y Commodore 64, Código Máquina.



Núm. 7
Analizamos el Generador de Sonido. Aplicaciones matemáticas con el ordenador. La memoria de video. Trucos, noticias.



Núm. 9
Características técnicas del Compact Disc. Tratamiento de datos. Test: Quick Disk. Trucos, libros, noticias, programas.



Núm. 3
Los joysticks, 256 caracteres programables, Z80 corazón de león, compro/vendo/cambio.

PARA HACER SU PEDIDO, RELLENE ESTE CUPON, HOY MISMO Y ENVIELO A MSX MAGAZINE BRAVO MURILLO, 377. Tel. 733 96 62 - 28020 MADRID

Ruego me envíen los siguientes números atrasados ....

al precio de **300** ptas. cada uno. Cuyo importe abonaré:

☐ POR CHEQUE ☐ CONTRA REEMBOLSO ☐ CON MI TARJETA DE CREDITO

☐ AMERICAN EXPRESS ☐ VISA ☐ INTERBANK

Número de mi tarjeta ....

Fecha de caducidad ....

NOMBRE ....

DIRECCION ....

POBLACION .... C.P. ....

PROVINCIA ....

FIGURA 7a

```
10 INPUT N
20 I=1
30 C=i*I
40 CB=C*I
50 R=SQR(I)
60 RC=I↑(1/3)
70 PRINT "I= "; I
80 PRINT "CUADRADO="; C
90 PRINT "CUBO="; CB
100 PRINT "RAIZ CUAD.="; R
110 PRINT "RAIZ CUBICA="; RC
120 PRINT
130 IF I=N THEN END
140 I=I+1
150 GOTO 30
```

FIGURA 7b

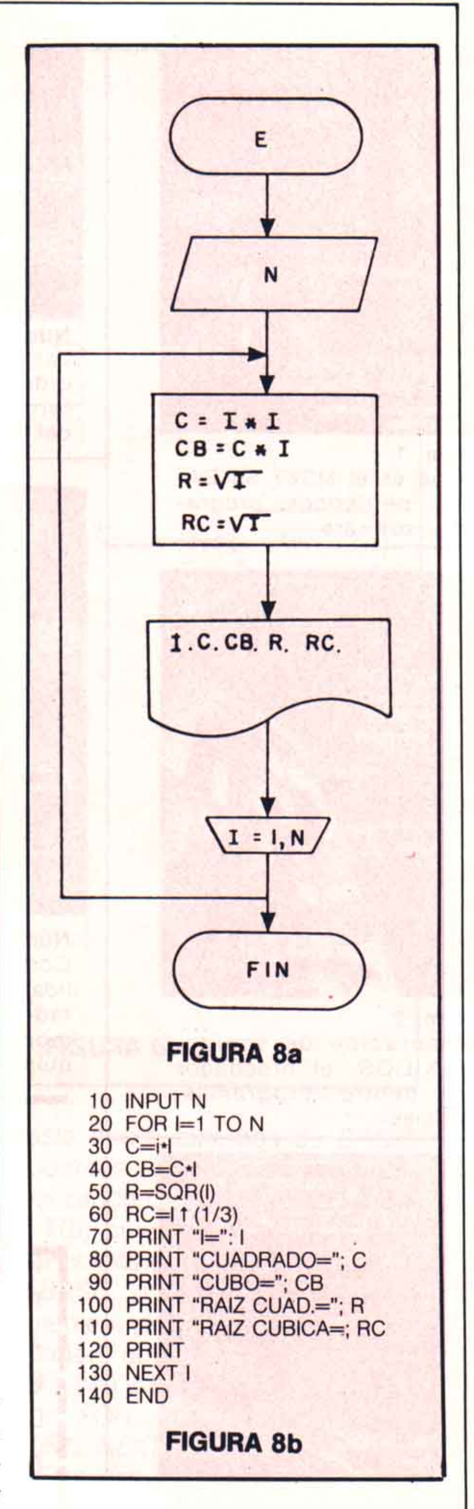


FIGURA 8a

```
10 INPUT N
20 FOR I=1 TO N
30 C=i*I
40 CB=C*I
50 R=SQR(I)
60 RC=I↑(1/3)
70 PRINT "I=": I
80 PRINT "CUADRADO="; C
90 PRINT "CUBO="; CB
100 PRINT "RAIZ CUAD.="; R
110 PRINT "RAIZ CUBICA=; RC
120 PRINT
130 NEXT I
140 END
```

FIGURA 8b

```
100 NEXT I
```

La línea 10 indica que para los valores de I comprendidos entre A1 y A2 (ambos incluidos) tomados de P en P se haga lo siguiente hasta que se encuentre la instrucción *NEXT I*. Así, el programa:

```
10 FOR I = 1 TO 10 STEP 2
20 PRINT I * I
30 NEXT I
40 ...
```

escribe los cuadrados de los números 1, 3, 5, 7 y 9, ya que Id empieza valiendo 1, después vale 1 + 2 = 3, a continuación 3 + 2 = 5, y así hasta que I = 9. Como el siguiente valor de I (9 + 2 = 11) está fuera del rango I = 1 TO 10, el cuadrado de 11 no se escribirá, y el programa continuará en la línea 40. Debes tener presente que I varía efectivamente, tomando los valores 1, 3, 5, 7 y 9.

```
10 REM LISTIN TELEFONICO - 1
20 REM =====================
30 REM
40 CLEAR 20000
50 SCREEN 0,,0
60 DIM NB$(100),DI$(100),TE!(100)
70 I%=0
80 I%=I%+1
90 PRINT "FICHA Nº";I%
100 PRINT
110 INPUT "NOMBRE";NB$(I%)
120 PRINT
130 INPUT "DIRECCION";DI$(I%)
140 PRINT
150 INPUT "TELEFONO";TE!(I%)
160 PRINT
170 PRINT
180 INPUT "¿DESEAS INTRODUCIR OTRO NUMER
     O...S/N";A$
190 CLS
200 IF A$="S" THEN 80
210 PRINT "BUSQUEDA DE FICHA"
220 PRINT
230 INPUT "DIME UN NOMBRE";NB$
240 FOR I1%=1 TO I%
250 IF NB$<>NB$(I1%) THEN NEXT I1%
260 CLS
270 PRINT "FICHA Nº";I1%
280 PRINT
290 PRINT "NOMBRE: ";NB$(I1%)
300 PRINT
310 PRINT "DIRECCION: ";DI$(I1%)
320 PRINT
330 PRINT "TELEFONO: ";TE!(I1%)
340 PRINT
350 PRINT
360 PRINT
370 GOTO 230
```

verás que te aparece el mensaje:

Out of string space

que quiere decir que has sobrepasado el espacio reservado para cadenas de caracteres. La instrucción *CLEAR* sirve para ampliar ese espacio de forma que:

CLEAR n

donde n puede valer desde 0 hasta el máximo de octetos de que nos permite disponer nuestro ordenador, a expensas de la memoria de programas y de la capacidad de almacenamiento de variables numéricas.

Por tanto, la línea 40 nos reserva espacio para 20.000 caracteres alfanuméricos. Como nuestra agenda va a tener 100 nombres, podemos emplear 200 caracteres en el nombre y la dirección de cada uno de nuestros amigos.

(La instrucción *CLEAR* sirve también para reservar memoria para rutinas en código máquina, como veremos más adelante).

El programa empieza actualizando un contador (línea 70) que nos indicará el número de la tarjeta que estemos manejando. Es decir, ese contador nos dirá hasta que subíndice, de los 100 que hemos reservado, hemos utilizado.

A continuación, el contador se aumenta en uno y comenzamos a introducir variables desde el teclado. Observa que el ordenador nos pregunta el nombre; y el que le demos nosotros lo asigna a la variable *NB$*(I%). Es decir, y en esto será la utilidad de las variables con subíndice, no necesitamos repetir el trozo de programa que comprende las líneas 90 a 170 más que una vez, dejando los subíndices de las variables indicados mediante el uso de otra variable. Por tanto, en *NB$*(I%), I% va variando de la forma que hayamos definido (línea 80) y no toma un valor constante: Ahora I%=1 pero cuando volvamos a pasar por la línea 80, I%=2, después I%=3,... y estaremos dando valores a las variables *NB$*(1), *NB$*(2), *NB$*(3),...

Una vez hemos introducido nombre, dirección y teléfono el ordenador nos pregunta si queremos introducir otra tarjeta (línea 80). Si contestamos S (mayúscula) el control vuelve a la línea 80, aumenta el valor de I%, etc. Si no contestamos S, el programa continúa.

A partir de la línea 200 podemos buscar las direcciones y números de teléfono de nuestros amigos. En la línea 230 el programa nos pide de un nombre, al que asigna la variable *NB$*. En las líneas 240 y 250 compara *NB$* con todos los *NB*(I) que tiene almacenado y cuando *NB$* es igual al *NB*(I) correspon-

```
10 REM LISTIN TELEFONICO - 2
20 REM =====================
30 REM
40 CLEAR 20000
50 SCREEN 0,,0
60 DIM A$(100,2)
70 B$(0)="NOMBRE    "
80 B$(1)="DIRECCION"
90 B$(2)="TELEFONO "
100 I%=0
110 I%=I%+1
120 PRINT "FICHA Nº";I%
130 FOR I1%=0 TO 2
140 PRINT
150 PRINT B$(I1%);
160 INPUT A$(I%,I1%)
170 NEXT I1%
180 PRINT
190 PRINT
200 INPUT "¿DESEAS INTRODUCIR OTRO NUMER
     O...S/N";A$
210 CLS
220 IF A$="S" THEN 110
230 PRINT "BUSQUEDA DE FICHA"
240 PRINT
250 INPUT "DIME UN NOMBRE";A$
260 FOR I1%=1 TO I%
270 IF A$<>A$(I1%,0) THEN NEXT I1%
280 CLS
290 FOR I2%=0 TO 2
300 PRINT
310 PRINT B$(I2%);": ";A$(I1%,I2%)
320 NEXT I2%
330 PRINT
340 PRINT
350 PRINT
360 GOTO 250
```



diente, salta a la línea 260 y nos escribe en la pantalla la ficha de nuestro amigo.

El programas 2 (al que le corresponde el diagrama 2. Ya dijimos que no interesa hacer un diagrama con excesivo detalle) es similar al programa 1; sin embargo, vemos en la línea 60 que hemos sustituido las variables dimensionadas *NB$*(100) *DI$*(100) y *TE!*(100) por una sola variable *A$*(100,2).

Esta es una variable con dos subíndices. *BASIC* nos permite dar a una variable tantos subíndices como nos permita la longitud de una línea. Así tenemos 100 variables que corresponden al valor 0 del segundo subíndice *A$*(I%,1) y *TE!*(I%) por *A$*(I%,2). Esto nos permite reestructurar nuestro programa según vemos en las líneas siguientes: En las líneas 70 a 90 damos a la variable con subíndice *B$*(I%) los «valores» NOMBRE, DIRECCION Y TELEFONO.

Observa dos cosas: No hemos dimensionado la variable *B$*(I%). *BASIC* MSX permite no dimensionar una variable con un subíndice si éste es menor o igual a 10. Por otra parte el primer índice de una variable con subíndice es siempre 0 para *BASIC* MSX, por lo que hemos tomado 0,1 y 2 en lugar de 1,2 y 3 para el subíndice de *B$*(I%) y para el segundo subíndice de *A$*(I%,J%).

Con los cambios que hemos hecho, podemos sustituir las líneas 100 a 150 del programa 1 por las líneas 130 a 170 del programa 2 y las líneas 270 a 330 por las líneas 290 a 320 del nuevo programa.

Estos cambios nos permitirán agilizar la edición de nuestro programa, según veremos en los próximos números.

J. Antonio Feberero

CURSO DE INGLES

The Gruneberg Linkword Language System es un sistema, para enseñanza de idiomas, más rápido y fácil que los métodos convencionales aplicados actualmente.
En poco tiempo, máximo 20 horas, te enseñará un vocabulario de 400 palabras y adquirirás unas buenas nociones de gramática. Esto te permitirá entender y ser entendido en tus viajes a lugares de habla inglesa o en tus contactos con personas que se expresen en ese idioma.
Por otra parte, el Sistema PlusData, consigue que el ordenador se convierta en un perfecto profesor que te explicará, orientará y corregirá, manteniendo en todo momento un "diálogo" interactivo de resultados sorprendentes.



–L. Taylor. "POPULAR COMPUTER WORLD":
"Quedé francamente atónito al comprobar la efectividad de la sugestión de imágenes como elemento de ayuda a la retención..."

–"PERSONAL COMPUTER WORLD":
"Un suceso fuera de serie..."

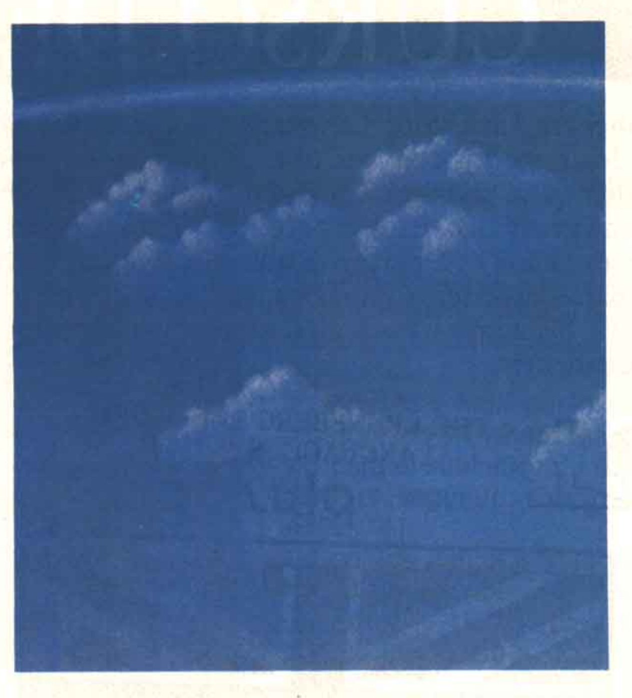
–Bill Barnet. "COMPUTER CHOICE":
"De todos los paquetes para aprender idiomas éste es el más interesante..."



Programas de EAO para EGB.
Cursos de Basic, Cobol, etc. AUTODIDACTAS.

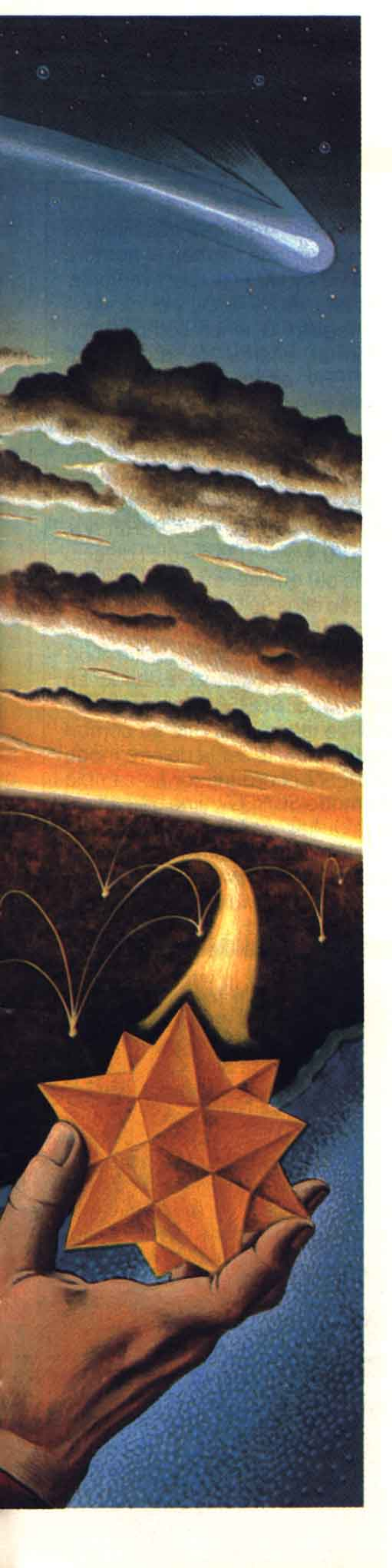
Nombre ....
Apellidos ....
Dirección ....
Población ....
D.P. .... Tlno. ....
Forma de pago: Reembolso ☐ Giro postal ☐ Envío talón ☐

☐ Curso de Inglés 1.ª parte. 10 lecciones Linkword. (Cinta) P.V.P. 6.900.-Ptas.
☐ Curso de Inglés 1.ª parte. 10 lecciones Linkword. (3,5"-Disk) P.V.P. 7.900.-Ptas.
ENVIAR ESTE CUPON A: PLUS DATA, S.A. C/. GRAN VIA, 661 pral. 08010-Barcelona. Tel. 246 02 02



La memoria de vídeo (VII)





Tras finalizar la explicación general del funcionamiento de los 16384 octetos que componen la memoria de vídeo, este artículo profundiza en el tema, analizando unos registros que complementan, e incluso en algunos casos controlan, a las tablas de la memoria de vídeo.

Los registros VDP

La memoria de vídeo, objeto de nuestro estudio, no sólo se compone de 16 K de memoria, sino que además posee 9 octetos aparte, llamados «registros *VDP*». Su misión es, por una parte, informar al programador de más cosas que suceden en la pantalla (aparte de la información que el programador puede lograr con los 16384 octetos); y por otra parte, alterar en cierta manera la pantalla, lo que sucede cuando estos registros son alterados por el programador. Esta segunda misión sólo pueden llevarla a cabo los primeros 8 registros *VDP* (numerados del 0 al 7), pues el nove-

no (*VDP* número 8) solamente sirve para la primera misión.

Los registros *VDP,* como cualquier otro octeto, admiten un número entero decimal del 0 al 255, o un número binario del &B0 al &B11111111, o un número hexadecimal del &H0 al &HFF. Para leer el número contenido dentro de un *VDP,* cosa que podréis hacer en todos los registros *VDP,* tendréis que ejecutar la orden:

PRINT VDP (n.° de registro)

y para introducir un número en un octeto, cosa que sólo podréis hacer con los registros del 0 al 7, tendréis que poner:

VDP (n.° de registro) = número a introducir

Los registros del 0 al 7

El registro 0, el primero de todos, es más fácil de comprender con el sistema de numeración binaria. A la primera cifra binaria por la derecha se le suele llamar *«EV»,* y en vuestro ordenador MSX lo más posible es que sea 0. Si lo ponéis a 1, sucederán cosas extrañas en la pantalla. Esta cifra binaria, este *«bit»,* tiene como misión admitir datos de otras memorias de vídeo. Admite datos si está a uno, y no los admite si está a cero.

La segunda cifra binaria por la derecha recibe el nombre de «M3», y será mejor que la expliquemos junto a los *bits* «M1» y «M2» que se hallan en el registro número 1.

Las seis restantes deben permanecer a 0. No tienen misión alguna, a diferencia de las *«EV»* y *«M3».*

El registro número 1, al igual que el anterior, será mejor tratarlo en el sistema de numeración binaria. La primera cifra binaria por la derecha, el primer *«bit»,* recibe el nombre de *«MAG»,* procedente de la palabra inglesa *«MAGNIFICATION»,* que significa «ampliación». Si recordáis, en el número que dedicamos a las figuras móviles, hablábamos de que el hecho de que los *pixels* de los *SPRITES* estén ampliados o no, no se podía determinar en las tablas, sino en otro lugar de la memoria de vídeo, e incluso hacíamos referencia a los registros *VDP,* señalando que ya hablaríamos de ellos cuando fuera oportuno. Pues bien, este es el momento: la primera cifra en binario del registro 1, si está a 0, implica que las figuras móviles que aparezcan en la pantalla serán de *pixels* sin ampliar, y si está a 1, implica que las figuras móviles que aparezcan estarán compuestas de *«pixels* ampliados».

La segunda cifra por la derecha recibe el nombre de *«SIZE»,* palabra inglesa que significa «tamaño», y que también se refiere a las figuras móviles. Si este *bit* está a 0, las figuras móviles que aparezcan serán de 8 filas x 8 columnas de *pixels,* y si está a 1, los *sprites* serán de 16 x 16 pixels.

Gracias a estos dos *bits,* ya es posible operar con figuras móviles sin recurrir a la setencia *SCREEN,* a la que el programador estaba antes restringido para poder especificar el tipo de figuras móviles que deseaba. Con estos dos *bits,* ya es posible controlar totalmente las figuras móviles desde la memoria de vídeo: los patrones se podían almacenar en unas tablas, los datos de la aparición en pantalla en otras tablas, y en los registros *VDP* su tamaño y si sus *pixels* están ampliados o no.

El tercer *bit* por la derecha no recibe nombre alguno y, como los seis *bits* del registro 0, no tiene ninguna misión encomendada y debe permanecer a 0.

La cuarta y quinta cifras binarias por la derecha reciben el nombre de *«M2»* y *«M1»* respectivamente. Los *bits «M1», «M2»,* y el *bit «M3»* del registro 0 tienen como misión informar acerca de qué modo *SCREEN* es el que hay en la pantalla en un momento dado. Si es el modo *SCREEN 0,* M1 = 1, M2 = 0, M3 = 0; si es el modo *SCREEN 1,* M1 = 0, M2 = 0, M3 = 0; si es el modo *SCREEN 2,* M1 = 0, M2 = 0, M3 = 1; y si es el modo *SCREEN 3,* M1 = 0, M2 = 1, M3 = 0. El defecto principal que tienen estos tres bits es que ellos no cambian de modo, solamente informan al programador de en qué modo se halla el ordenador. Si cuando, estando en un modo *SCREEN,* situáis esos *bits* de tal manera que su combinación resulte ser la de otro modo *SCREEN,* el ordenador no cambia de modo *SCREEN,* sino que adopta algunas características de un modo y otras de otro. Podéis comprobar en vuestro ordenador MSX cómo, en este caso, comienza a



comportarse de una manera extraña.

La sexta cifra binaria por la derecha recibe el nombre de *«IE»*, y es la encargada de activar o desactivar la salida de interrupción de la memoria de vídeo. Normalmente ese *bit* está a 1 (activada), pero si lo ponéis a 0 sin estar dentro de un programa, podréis comprobar cómo desobedece todas las órdenes que le deis desde el teclado, hasta tal punto que no pintará absolutamente nada ni hará ninguna otra cosa hasta que no apaguéis el ordenador. Sin embargo, si estáis dentro de un programa, el programa seguirá ejecutándose, pero sin que podáis cortarlo con *CTRL/STOP*, ni podáis hacer nada, pues habrá inutilizado el teclado. La única manera de que vuelva a hacer caso al teclado es incluir en el programa una orden que restituya en el *bit IE* el 1.

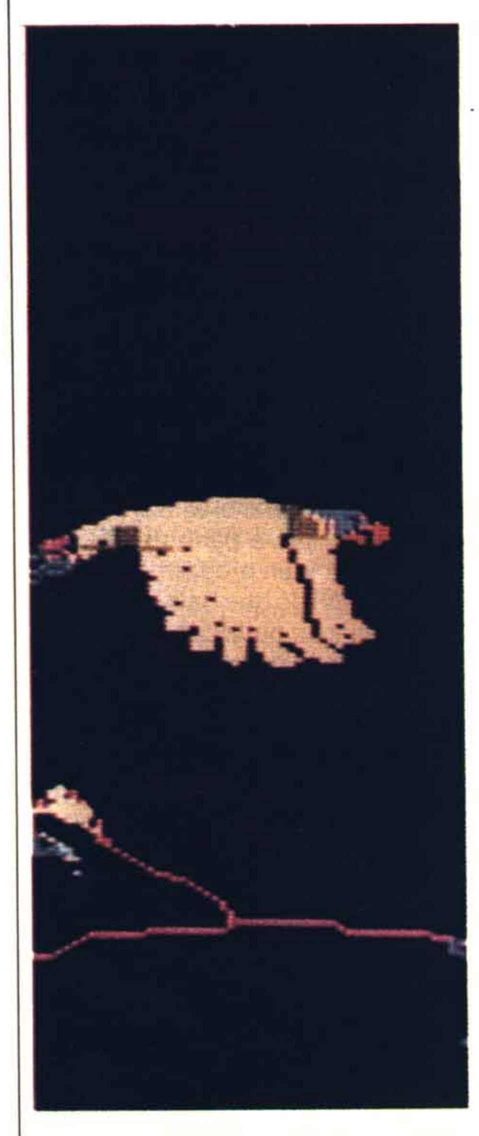
La séptima cifra binaria por la derecha recibe el nombre de *BLANK*, y su misión es desconectar la pantalla, de tal manera que el programador no pueda ver absolutamente nada en ella. El *bit IE*, al ponerse a 0, no hacía caso al teclado, pero el observador podía ver en pantalla lo último que hubiera escrito. Sin embargo, al ponerse a *O BLANK*, desaparece todo lo escrito, y la pantalla se queda con una claridad difusa teñida del color de borde. A diferencia del *bit* anterior, *BLANK* sí obedece al teclado, y aunque no podáis verlo, lo que vosotros tecleáis en el ordenador se va almacenando en la pantalla, y si tecleáis órdenes, el ordenador las obedece, aunque vosotros no podáis ver ni vuestras órdenes, ni el «OK» que da como respuesta. (No siempre da «OK» pues, en estos casos, al escribir órdenes a ciegas, la respuesta más común es *«Syntax error»* o *«Type mismatch»*, dado que lo más posible es que las órdenes estén mal escritas.) A ciegas aún, podéis reescribir que el *bit BLANK* vuelva a 1, su estado normal, y al obedecer el ordenador, la pantalla se conectará y podréis ver todo lo que habéis escrito.



El octavo y último *bit* del registro 1 se llama *«VRAM»*, y su misión es seleccionar el tipo de memoria de vídeo a utilizar. El tipo de memoria de vídeo que estamos explicando en esta serie de artículos es la de 16K, la más común en los ordenadores MSX, y que se establece poniendo a 1 el *bit VRAM*. Poniéndolo a 0 establece una memoria de vídeo de 4K, que sólo poseen algunos modelos MSX muy escasos.

Los registros *VDP* 2, 3, 4, 5 y 6, que son los que veremos a continuación, tienen como misión indicar en qué octeto de los 16384 octetos de la memoria de vídeo empiezan ciertas tablas, en cada uno de los sistemas de almacenamiento de los diversos modos *SCREEN*. Estos registros ciertamente controlan a las tablas, en este sentido, pero no las controlan totalmente, pues cuando fijan el comienzo de una tabla en un octeto concreto, esa tabla habrá sido trasladada a ese octeto en algunas cosas, pero seguirá donde estaba en otras cosas. Vamos a poner un ejemplo: si la tabla 0 del sistema de almacenamiento del modo *SCREEN 0* la trasladamos de donde está (el octeto 0) al octeto 1024, lo que podréis ver en la pantalla son los caracteres cuyos códigos se hallan a partir del 1024, pero si desde un programa ejecutáis un *PRINT*, lo que haya que pintar se pintará a partir del octeto 0, o sea, la antigua posición de la tabla 0; también si tecleáis algo, lo tecleado se introducirá en la memoria de vídeo a partir del octeto 0, y no a partir del 1024. Esto tiene la ventaja de que podéis ir preparando una pantalla a partir del octeto 0, a la vez que el observador está leyendo la pantalla que antes habéis preparado a partir del 1024, gracias a la instrucción *VPOKE*.

El registro 2, al igual que los registros 3, 4, 5 y 6, es mejor tratarlo en decimal. Este registro marca el octeto en el que comienzan las tablas 0 en el modo *SCREEN 0*, 5 en el modo *SCREEN 1*, 10 en el modo *SCREEN 2* y 15 en el modo *SCREEN 3*. Pero estas tablas no pueden comenzar en un octeto cualquiera, sino en un octeto múltiplo de 1024, siendo el número entero correspondiente a esa multiplicidad el que hay que introducir en el registro. Por ejemplo, si queréis que sea del sistema de almacenamiento que comience en el octeto 5120, teniendo en cuenta que 1024 x 5 = 5120, *VDP* (2) ha de tomar el valor de 5. El valor máximo que debe tomar el *VDP* nú-



mero 2 es 15, pues si es 16 el ordenador entiende que debe situar en 0 la tabla correspondiente. Introduciendo 17, el ordenador actúa como si se hubiera introducido 1, con 18 el 2, ... y con 31 el 15. Y con 32 el 0 de nuevo, y vuelta a empezar, hasta llegar a 255 donde, como es lógico, el ordenador actúa como si se hubiera introducido 15.

Habéis de tener mucho cuidado con esta traslación parcial de tablas, pues puede darse el caso de que se sitúen 2 o más tablas en una misma zona de memoria, y al direccionar los octetos de esa zona para lograr unos resultados en una tabla, este cambio de octetos traiga consecuencias inesperadas en las otras tablas.

El registro 3 controla la tabla 6 del sistema de almacenamiento del modo *SCREEN 1,* con lo que esta tabla se puede situar en cualquier octeto múltiplo de 64, y al igual que en el registro *VDP 2,* el valor de *VDP* (3) ha de ser el factor de multiplicidad, que normalmente es 128, pues la tabla 6 está situada a partir del octeto 8192 (128 x 64 = 8192).

El registro 4 contola las tablas: 2 del modo *SCREEN 0;* 7, del modo *SCREEN 1* y 17 del modo *SCREEN 3.* El primer octeto de la tabla ha de ser múltiplo de 2048 y, como siempre, el valor a introducir en el registro *VDP* es el factor de multiplicidad.

El registro 5 controla las tablas: 8 del modo *SCREEN 1,* 13 del modo *SCREEN 2* y 18 del modo *SCREEN 3.* En este caso, el octeto inicial ha de ser múltiplo de 128 y, de nuevo, el número a introducir en el registro es el factor de multiplicidad.

Por último, el registro 6 controla las tablas: 9 del modo *SCREEN 1,* 14 del modo *SCREEN 2* y 19 del modo *SCREEN 3.* En este caso, lo mismo que sucede en el registro 4, el octeto inicial ha de ser múltiplo de 2048.

El registro 7 se ocupa de una cuestión fundamental en el tratamiento del color: los colores de fondo y de tinta «fijos» que adopta el ordenador. (NOTA: si recordáis, hay maneras de introducir colores de tinta y de fondo distintos de los colores ya establecidos con una sentencia *COLOR,* gracias a las tablas 6, 11 y 17 en los sistemas de almacenamiento de los modos *SCREEN* 1, 2 y 3, repectivamente.) Cuando, con una sentencia *COLOR,* ordenáis un color de tinta y un color de fondo concretos, éstos se almacenan en el registro 7. Por ello conviene tratar este octeto en hexadecimal: la cifra hexadecimal de la izquierda es el color de la tin-



ta, y la cifra de la derecha es el color de fondo. (La lista con los 16 colores MSX y sus códigos en hexadecimal la publicamos en el número 8 de **MSX Magazine,** en el artículo «La memoria de vídeo (II): caracteres en color».)

El registro número 8

El registro *VDP* número 8, que sólo es posible usarlo para leer, es muy útil para el ordenador, pero menos útil para el programador. Será mejor tratarlo en binario pues, al igual que el registro *VDP* número 1, sus funciones se distribuyen *bit* por *bit,* es decir, cifra en binario por cifra en binario. El primer *bit* empezando por la izquierda se llama «F», aunque también recibe el nombre de «*flag* de interrupción». Un «*flag*» es una cifra en binario, un *bit* cuya misión es puramente arbitraria, y en la que los valores que puede tomar, 0 y 1, cobran un muy importante significado. Es una manera muy hábil de concentrar la mayor cantidad de información en la mínima cantidad de memoria. Los ordenadores suelen usar mucho estos «*flags*», cuyo significado en español es «banderas», y que se llaman así casi metafóricamente, pues si al ordenador le sucede una cosa concreta, se «levanta» la «bandera», es decir, el «*flag*» toma el valor de 1, y si no le sucede, se «baja» la «bandera», o sea, el «*flag*» toma el valor de 0. En el caso concreto del *bit F,* se pone a 1 cuando ha terminado el análisis de la pantalla, y se pone a 0 después de comprobar el registro *VDP* 8.

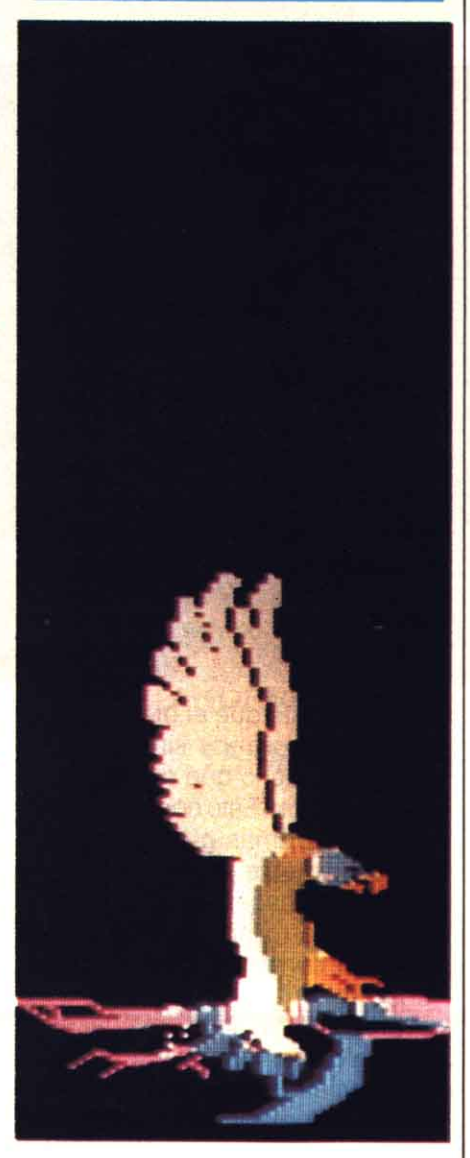
El segundo *bit* por la izquierda recibe el nombre de «5S», y se «levanta» su «bandera», es decir, se pone a 1, cuando más de cuatro figuras móviles se alinean horizontalmente. Mientras esto no suceda, su «bandera» permanece «bajada», o sea, se pone a 0. En general, este bit estará a 0 en todos los modos *SCREEN,* excepto en el modo *SCREEN 0,* donde no hay figuras móviles, y donde siempre está a 1.

El tercer *bit* por la izquierda se llama simplemente «*C*» (no se debe confundir este *bit* C con el registro C que se usa en Código Máquina, ni tampoco se debe confundir el *flag* F con el registro F del Código Máquina, pues a pesar de tener el mismo nombre, son cosas distintas) y vale 1 si dos o más figuras móviles se superponen sobre un mismo píxel. Mientras no se superpongan, este «*flag*» valdrá 0. En el *bit* C, precisamente, es en el que se basa la orden en *BASIC* «*ON SPRITE GOSUB*», que bifurca un programa en una dirección concreta en el caso de que este bit valga 1.

En los 5 *bits* restantes del registro 8 se halla el número de la figura móvil que se encuentra alineado horizontalmente con otras cuatro, provocando su parcial desaparición. Pero el número de cinco cifras binarias que se almacena no es el número del patrón de la figura móvil que se haya superpuesto, sino el número del plano de proyección donde se halla la figura móvil que sobra. Esto es lógico, pues habiendo 32 planos numerados del 0 aL 31, lo normal es que se necesiten 5 cifras binarias para expresar estos números (en el caso extremo, 31 = &B11111). Como es natural, estos 5 *bits* sólo se utilizan cuando hacen falta, es decir, cuando 5 figuras móviles se superponen, con lo cual el *bit* 5S se pone a 1 y estos cinco *bits* almacenan el plano del *sprite* que sobra.

Principales aplicaciones

No todo lo que hemos visto en este capítulo dedicado a los registros *VDP* tiene aplicaciones para el programador; por ejemplo, el *bit EV* no tiene apenas. Sin embargo, hay algunas cosas que son muy interesantes, como el *bit MAG,* gracias al cual se pueden alternar figuras móviles de *píxels* ampliados y sin ampliar en muy cortos intervalos de tiempo, lo que puede provocar unas sensación de confusión en la pantalla, como de explosión, muy útil en programas de



guerra, o de extraterrestres, si se aplica al tiempo un sonido de explosión adecuado. El *bit SIZE* puede emplearse sustituyendo a la orden *SCREEN* en su coordenada correspondiente a las figuras móviles. Los *bits M1, M2 y M3* pueden servir para que a lo largo del listado, el programa tome una decisión u otra, en función del modo *SCREEN* en el que se encuentre.

Para sustituir a un *ON STOP*, poner a 0 el *bit IE* puede ser muy útil, pues inutiliza el teclado y, por lo tanto, permite que el programa se vaya desarrollando sin depender para nada de lo que pueda hacer el observador. Esto conviene usarlo en programas o en subrutinas exclusivamente expositivas, cuya misión sea hacer largas operaciones o desarrollar cosas en la pantalla sin contar para nada con el observador.

El *bit BLANK* sirve para dar sensación de confusión, si se apaga y se enciende en poco tiempo. También es útil para desarrollar complicados dibujos cuyo trazado dure varios segundos, y que se pretenda que su aparición en pantalla sea instantánea. Es muy sencillo: se apaga el *bit BLANK*, se traza el dibujo, y se vuelve a poner a *1 BLANK*. Esto dará el efecto de una aparición instantánea, si durante los segundos previos entretenemos al observador con algún sonido, o alguna música que aparte su atención de la pantalla. También puede servir como culminación a una explosión, que puede comenzar con el *bit MAG* ampliando y disminuyendo una figura móvil de explosión, y puede terminar con la desaparición de la pantalla (ahí interviene el *bit BLANK*).



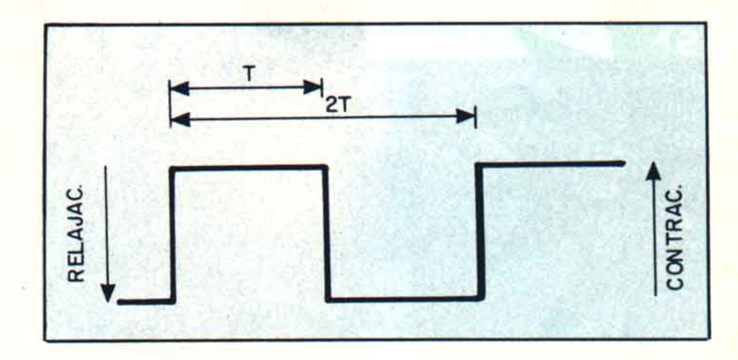
Por último, poder controlar las tablas por medio de los registros *VDP* del 2 al 6, puede llegar a ser una de las ayudas más importantes que el programador puede emplear en la memoria de vídeo. Os pondremos unos cuantos ejemplos a continuación:

— Si desplazáis la tabla 0 a lugares diferentes de la memoria de vídeo, y en cada uno de esos lugares habéis creado una pantalla con un texto concreto, y a lo largo del programa tenéis que volver varias veces a esas pantallas, no tendréis que recurrir a una *CLS*, ni tendréis que volver cada vez que lo necesitéis a la subrutina correspondiente con una larga serie de *PRINT*, sino que con darle valores diferentes al registro *VDP* (2), aparecerán las diferentes pantallas. Esto os ahorrará tiempo y memoria. En la tabla 17 se pueden desarrollar varias pantallas del modo *SCREEN 3*, de tal manera que si dos pantallas son muy parecidas y sólo se diferencian en que, por ejemplo, un dibujo grande tiene una posición algo distinta (ese dibujo grande puede ser un objeto, una nave nodriza, o algo por el estilo), al alternar ambas pantallas da la impresión de que está temblando, o de que se está moviendo. Esto no se puede lograr por medio de figuras móviles, pues tienen un tamaño limitado (las más grandes son de 32 x 32 *píxels*), y sin embargo de esta manera podéis lograr que se muevan objetos mucho más grandes.

— Desplazando la tabla 6, podéis lograr que un mismo texto tome diferentes colores, lo que realizado muy rápidamente puede convertirse en un efecto *«FLASH»* a gran escala.

Y si habéis leído hasta aquí, toda la memoria de vídeo está a vuestro alcance: ya tenéis conocimientos suficientes como para lograr en vuestros programas efectos muy importantes para el programador. En el próximo y último capítulo, nos referiremos al *BIOS* (*BASIC INPUT-OUTPUT SYSTEM*), y a cómo controlar la memoria de vídeo desde el Código Máquina.

Síntesis de voz para los MSX (I)





Uno de los principales problemas a la hora de diseñar un ordenador es el estudio del sistema de comunicación que el ordenador utilizará para poder entenderse con el usuario. Este problema queda resuelto con el acoplamiento al ordenador de una pantalla de forma que podamos leer en ésta todos los datos que la máquina nos da. Ahora bien, llegado un momento en que la tecnología es lo suficientemente avanzada, podemos hacer que el ordenador sea capaz de emitir palabras tal y como lo hacemos nosotros, haciendo la comunicación de una forma más fácil y cómoda. Esto es justamente lo que vamos ha hacer con nuestro MSX.

Existen muchos y complicados métodos para hacer que el ordenador hable, pero tienen en su mayoría el inconveniente de que hay que acoplar a la máquina una serie de circuitos (convertirodes analógico/digital y digital/analógico) de complicada construcción que encarece y haría dificultosa la tarea. No obstante, existe un método poco complejo por el cual, sin tener que acoplar ningún tipo de interface, podrá hacer que nuestro MSX nos diga frases como «HOLA», «INTRODUZCA UN NUMERO», etc., haciendo que el ordenador nos satisfaga más todavía.

Posiblemente el lector estará pensando en el *chip* de sonido como el encargado de hacer esta delicia, pero esto no es cierto. El *chip* de sonido es perfecto para ejecutar música pero el sistema que nos interesa deberá ser de tal forma que sea capaz de realizar cualquier tipo de onda, y no unos formatos ya establecidos. Usaremos el port de cassette y el port que controla el *«clic»* del altavoz para SINTETIZAR la voz.

Estos *ports* son lo suficientemente interesantes como para realizar un estudio completo de ellos por lo que además de presentar el programa «volcals», capaz de emitir frases habladas, explicaremos de qué forma se pue-



de sintetizar sonido con el uso de dichos *ports*. Pasemos a estudiar el port que gobierna el altavoz y de qué forma este *port* llega a hacer hablar al MSX.

El port ABH

Este puerto es el que se encarga de la programación de la *PPI* pero además a través de él podemos hacer que la membrana del altavoz del televisor se contraiga o se relaje. Así, *OUT &HAB,15* hará que la membrana del altavoz se contraiga y *OUT &HAB,14* que se relaje. Esta pareja de instrucciones es la que realiza el ordenador cada vez que pulsamos una tecla produciéndose así el «clic» ya conocido por nosotros. Si ejecutamos la primera instrucción escucharemos un *«clic»* ya que el altavoz que antes estaba relajado se contraerá y al pulsar después cualquier tecla haremos que dicha menbrana se relaje, lo que significa que la segunda instrucción no hará nada que podamos notar ya que relajaría una membrana que se encuentra en reposo. Ahora bien, si en un programa *BASIC* hiciéramos ejecutar ambas instrucciones unas tras otras, no haríamos otra cosa más que mover la membrana del altavoz, es decir, crearíamos un sonido.

```
10 OUT &HAB,15:REM CONTRAE MEMBRANA
20 OUT &HAB,14:REM RELAJA MEMBRANA
30 GOTO 10:REM HAZLO OTRA VEZ
```

podemos incluso avanzar un poco más; si entre la línea 10 y 20 colocásemos un contador o algo que hiciese retardar la velocidad de los movimientos, seríamos capaces de variar la frecuencia del sonido emitido:

```
 5 INPUT "INTRODUCE UN NUMERO";A
10 OUT &HAB,15
15 FOR I=1 TO A: NEXT I
20 OUT &HAB,14
25 FOR I=1 TO A: NEXT I
30 GOTO 10
```

Ahora, según el valor que introduzcamos en A, así será la frecuencia del sonido. En la figura 1 tenemos de forma esquemática el sonido generado.

Un generador de frecuencias de audio

Si pudiéramos saber exactamente cuánto tiempo tarda cada instrucción del programa anterior, mediante un simple cálculo, podemos generar un sonido con la frecuencia que quisiéramos, pero esto en *BASIC* es prácticamente imposible. Una instrucción *BASIC* no tarda en ejecutarse siempre un mismo tiempo, esto depende de muchos factores, y aunque la variación no fuese grande, ya no podríamos usar el ordenador como generador de frecuencias de buena precisión. Este problema lo solucionaremos haciendo el programa en código máquina.

Para aquellos lectores que no entiendan o no les interese el código les será interesante conocer que dicho programa realiza justamente lo mismo que el *BASIC* anterior pero con dos ventajas:

1. En código máquina podemos conocer cuánto tarda exactamente una instrucción en ejecutarse.
2. En código máquina podemos hacer, mediante la inhibición de las llamadas interrupciones enmascarables, que una rutina dure siempre el mismo tiempo.
3. En el listado ensamblador que viene a continuación también se incluyen los llamados «estados de reloj» de las instrucciones que interesan. Estos datos son muy importantes para calcular el tiempo de ejecución.

```
          OGR 55002
          DI
INI:      LD HL,(55000)
          LD A,15
          OUT (=AB),A
          CALL CONT        17
          LD A,14
          OUT (=AB),A
          CALL CONT        17
          JP INI
CONT:     LD HL,(55000)    16
A1:       DES HL            6
          LD A,H            4
          OR L              4
          JP NZ,A1         10
          RET
```



Antes de ejecutar el programa, habremos introducido desde el *BASIC* en la dirección de memoria 55000 (y siguientes) un valor comprendido entre 0 y 65535 el cual es el homólogo del introducido en A en el programa *BASIC* anterior Al principio inhibiamos las interrupciones para evitar que el microprocesador se «despiste» de nuestro programa, contraemos el altavoz, pasamos a un bucle típico en el cual se realizarán tantos ciclos como sea el valor introducido en la dirección 55000, retornamos para relajar la membrana y volvemos a contar. Así sucesivamente hasta que pulsemos la tecla *RESET;* único sistema de parar el programa. Ahora sólo nos falta saber a qué frecuencia corresponde el valor introducido en la dirección 55000. Estos cálculos son algo tediosos, pero nada complejos.

Para calcular cuánto tiempo tarda en ejecutarse una instrucción en código máquina sólo hace falta multiplicar el número de estados de reloj correspondientes a cada instrucción (estos valores los podremos encontrar en cualquier libro especializado en el *Z-80*) por el inverso de la frecuencia de reloj de nuestro ordenador en ciclos por segundo. Como la frecuencia de reloj es de 3.5 MHz (3500000 ciclos por segundo) la instrucción *"CALL CONT"* durará 17*(1/3500000) segundos, o sea, 4.8 millonésimas de segundos (¡¡!!). De igual forma, la subrutina encargada de esperar tardará (N es el valor introducido en la dirección 55000).

$$\frac{17+16+10+N(6+4+4+10)}{3500000}$$

segundos

O lo que es lo mismo (T=tiempo en segundos)

$$T = \frac{434+24N}{3500000} \text{ segundos}$$

Como T es el tiempo que tarda media longitud de onda (figura 1), la frecuencia del sonido será (F= frecuencia en ciclos por segundo);

$$T = \frac{1}{2T} \text{ ciclos/segundo}$$

O lo que es igual;

$$F = \frac{1750000}{43+24N} \text{ ciclos/segundo}$$

El programa *BASIC* que viene a continuación es el encargado de introducir en memoria y ejecutar dicho programa máquina.

```
10 DI= 55002
20 FOR I=0 TO 30
30 READ J:POKE (DI+I),J
40 NEXT I
50 DEFUSR=55002
60 INPUT "INTRODUZCA
   NUMERO";A
70 B=INT (A/256):POKE
   55001,B
80 POKE 55000, ((A/256)–B)
   *256
90 X=USR (I)
100 DATA 243,42,216,214,62,15,
211,171,205,239,214,62,14,211,
171,205,239,214,195,214,42,216,
214,43,124,181,194,242,214,201
```

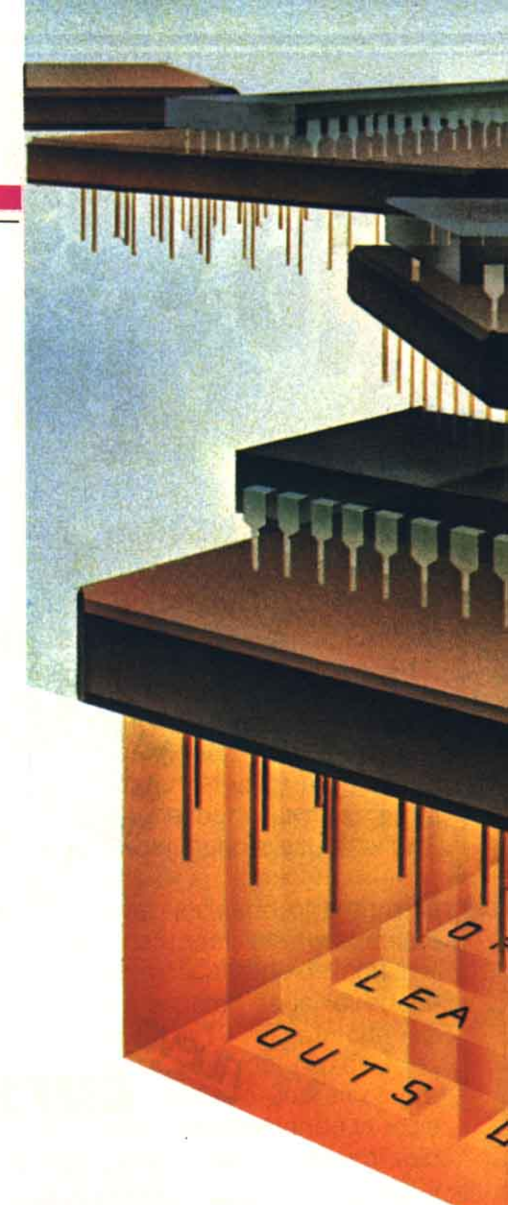
Desde la línea 10 hasta la 40 introducimos el programa en código, en 50 lo definimos y en 70 y 80 introducimos el valor preguntado en memoria para terminar en la línea 90 haciendo ejecutar el programa.

Una vez tecleado el programa *BASIC* cargador, nos preguntará el número encargado de indicar el tiempo a esperar, así, si introducimos el valor 1000, la frecuencia del sonido será de 72.78 ciclos por segundo. Sólo podremos parar el programa pulsando la tecla *RESET* pero por regla general sólo se borrará el programa *BASIC* y no el máquina por lo que únicamente se tendrá que volver a teclear desde las líneas 60 a la 90. De todas formas no sería mala idea grabar el programa entero en cassette o disco.

Aunque parezca que no, hemos realizado algo muy importante. De forma muy rudimentaria hemos digitalizado un sonido, hemos convertido dicho sonido en un número. En el próximo capítulo usaremos el *port* de cassette para que nos trasforme a números cualquier sonido que le introduzcamos, de forma que usando el *port* de altavoz podamos volver a reproducirlo. Así haremos hablar al MSX.

Juan Jiménez León

Utilidades de la RAM



Para el manejo de datos y para iniciarse en las aplicaciones de la programación ofrecemos una serie de temas dedicados especialmente a dos grupos de personas:

Los que disponen de un MSX y/o SVI de la serie 300 que desde el principio precisan de la práctica diáfana con ejemplos comentados y los que dudan en la clase de ordenadores con quien compartir los ratos más o menos de ocio, pero en plan más bien serio, aunque no hay que pasarse.

Los ejemplos pueden seguirse en cualquiera de los sistemas citados aunque es lógico que si no se dispone de ordenador propio o prestado los ejemplos prácticos puden estudiarse con ciertas limitaciones.

La manera más idónea de manejar correctamente un aparato tan complejo, es estar con él, «jugar con él», en un sentido amplio y por ello preveer una diversa actitud en los diferentes problemas que nos planteamos, con una buena dosis de paciencia, sobre todo al principio, para que todos podamos empezar con buen pie este camino.

De la cogida que se preste a curación, nada se suele recordar con el tiempo, pero menos si fue mortal. Así esperamos vuestras ideas y comentarios. Responderemos a todas las consultas salvo en las referentes a marcas de fabricación para el ordenador o periféricos más adecuados dentro del contexto de esta serie, cuya extensión la hacemos depender de vuestra aceptación. Cada artículo constituye un tema y todos los temas no tienen otra conexión que la memoria *RAM*.

El ordenador

Cuando no se poseen demasiados conocimientos sobre informática asaltan muchas dudas en la elección de nuestro futuro «amigo»; para que la amistad sea todo lo duradera que la técnica y nuestro bolsillo nos permitan deben tenerse en cuenta:

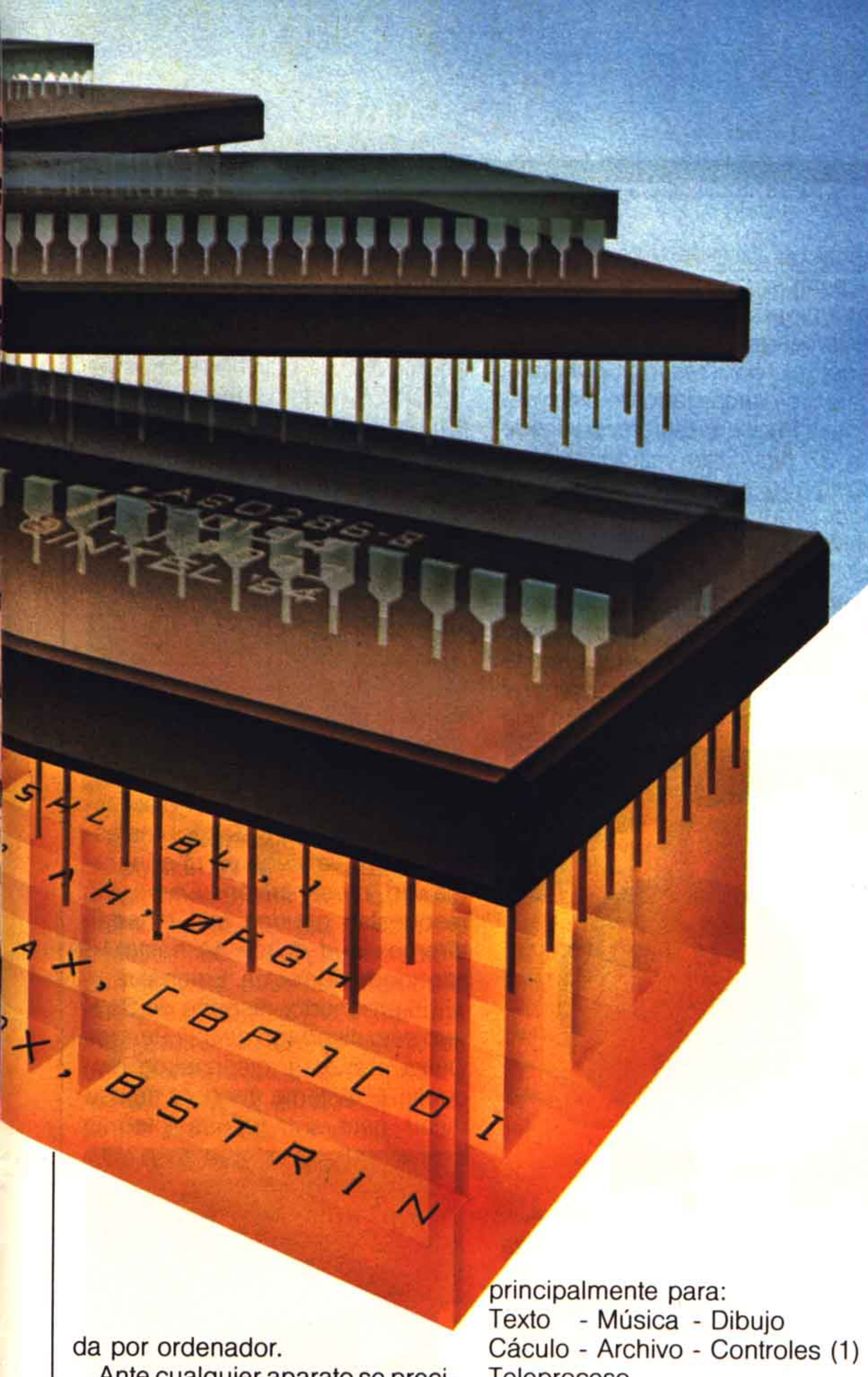
a) Principal utilidad de tarea.
b) Duración prevista en años.
c) Programas editados (software).
d) Servicio postventa.
e) Documentación adjunta.

No cabe duda que el propio vendedor debería ser la persona capacitada para solucionar cada caso, pero sabemos que por desconocimiento o compromisos no siempre se cumple esta posibilidad.

Las empresas distribuidoras o los mismos fabricantes son juez y parte, por lo que no suelen ser imparciales. Su información está «hinchada». Las revistas serias ilustran pero no pueden aconsejar con respecto a marcas por ética profesional.

Entonces, ¿qué? No existe otra alternativa que la investigación por parte del propio interesado. El manejo del ordenador (mini-, micro-, nano-, etc., son prefijos un tanto complejos y esperamos aclararlos más adelante) requiere una base, por cierto no difícil a pesar de su complejidad, para usar adecuadamente los medios de que dispongamos en una relación precio/usuario que sea la más conveniente.

No podemos dedicar estos métodos a los que tienen prisa ya que es muy mala consejera en estas lides ni tampoco a los que simplemente desean un juego sin más. Una faceta de la que nos ocuparemos convenientemente es la que se encarga de la enseñanza asisti-



da por ordenador.

Ante cualquier aparato se precisan conocimiento del proceso y manejo.

Si se pretende lavar de tres maneras, ¿para qué una lavadora de 30 programas? Esto sería un desconocimiento del proceso.

Si desconocemos para qué sirve el filtro o qué se persigue con el centrifugado no podemos emplear programas adecuados.

El ordenador puede utilizarse principalmente para:

Texto - Música - Dibujo
Cáculo - Archivo - Controles (1)
Teleproceso
Programación activa o pasiva
Didáctica (2)
Juegos: de acción / de mesa y tablero / de lógica y/o inteligencia

Supongamos que la música sea el principal objetivo de un usuario. Para ello se inclinará por aquel que facilite el tratamiento de solfeo, composición, edición, etc. Lo mismo se dirá respecto al texto y al dibujo. O sea, que las características generales de un ordenador no pueden resultar determinantes para la aplicación principal.

Además un ordenador no sólo se circunscribe a su unidad de control y memoria. Constará de una serie de periféricos que sean utilizados para esa función principal detrás de la cual nos vamos a inclinar preferentemente.

La principal distinción entre una máquina de calcular y un ordenador estriba en que este último tiene gran capacidad de almacenamiento de datos y la posibilidad de su reutilización, y renovación. Por ello al adquirir un ordenador es muy conveniente que se soliciten las medidas para que quepa en nuestro rincón, pero mucho más interesante serán sus posibilidades hacia las tareas que le vamos a encomendar.

Capacidad general

No deben confundirse la capacidad de almacenamiento en la memoria con el traslado de estos datos a uno o varios archivos.

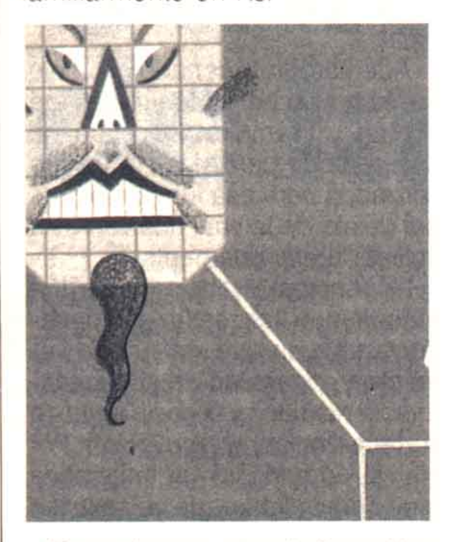
Un programa completo, cuya ejecución se prevea sin recurrir a medios periféricos ocupa la memoria *RAM.* Otro puede utilizar *RAM* y archivos simultáneamente e incluso se podrían usar sólo los archivos, aunque este caso para las utilizaciones medias que nosotros precisamos es muy poco corriente.

Necesitamos, pues, conocer la capacidad de que disponemos para obrar en consecuencia. Y esto no puede simplificarse ciñéndose exclusivamente a un número y ya está.

Si un piso se suele medir en metros cuadrados y las velocidades en kilómetros por hora, la ca-

(1) Aparatos, mecanismos y robots.
(2) Enseñanza asistida por ordenador.

pacidad de memoria de un ordenador se mide en *kilobytes*, o más familiarmente en Ks.



Recordemos que el dispositivo interno del ordenador sólo «entiende» Ø y 1, «hay» algo, significa para él un 1 y «no hay» debe asociarlo a un Ø. Cero que se raya diagonalmente para distinguirlo de la letra o.

Cada posición de si/no *(bit)* se asocia a otras siete formando un octeto *(byte)* para ganar velocidad en la transmisión interna. No es lo mismo enviar/recibir un *bit* detrás de otro que varios a la vez. En este caso ocho *bits* forman el envío de un *byte* cada vez. Como ejemplo diremos que cada letra:

E r Y n ñ Ñ w Q h l j

número:

Ø 4 1 8 5 6

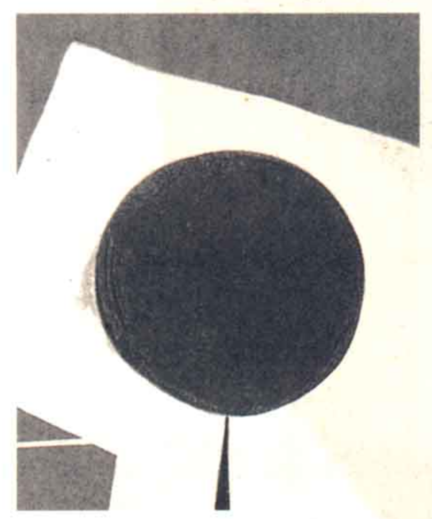
ó signo:

$ " / (&) ç %

ocupan aproximadamente un *byte* en la memoria y un espacio en la pantalla de ocho *bytes* (64 *bits*).

Por ello no es lo mismo utilizar un texto que un dibujo, números que variables alfanuméricas, bucles que circuitos contadores (A= A+1). Todos ellos disponen y utilizan distintas cantidades de memoria, es decir de *bytes*.

La capacidad de memoria viene determinada por la capacidad de posibilidades de direccionamiento del microprocesador, que es el «cerebro» encargado de este menester. Pueden establecer desde 8 a 32 *bits* cada vez. Mencionamos ahora el *Z-80* como microprocesador más común de 8 *bits* empleado en los ordenadores *MSX* y *SVI* que puede establecer y controlar una a una 65536 direcciones de ocho *bits* y por lo tanto 524.288 *bits*, lo cual nos permite aproximarnos a una idea general de los medios existentes, pero que



no concuerda con la que se utiliza que es grupos de 1024 *bytes* (K) es decir un *kilobyte*. Decimos, por ello, que 64 K cuando nos referimos a la memoria de un ordenador sin más especificaciones y esto no es exacto.

Un ordenador necesita un lenguaje de comunicación entre su entendedera (ceros y unos) y la nuestra. Uno de estos lenguajes es el *BASIC* que en el caso que nos ocupa necesita unas 32 K para su propio uso. Esta memoria es accesible pero no puede cambiarse. Y por fin vamos al grano. Disculpad el «rollo» inicial pero está dedicado a los indecisos. Era simplemente para que luego no entren en el grupo de los arrepentidos o midan un poco mejor antes de iniciar este fascinante mundo.

CONECTAR el ordenador si disponéis de él. Cuando tengáis ante la vista esa pantalla azulada o negra, pero expectante, ávida de vuestros impulsos, si no lo está actuar un *CLS*, escribir:

PRINT FRE (Ø)

y lógicamente *ENTER*.

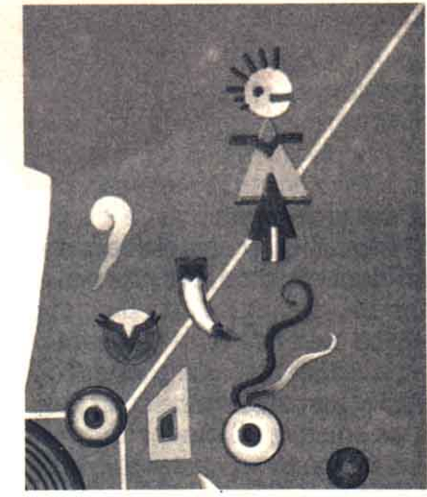
El número con que responde el ordenador son los *bytes* disponibles en la memoria *RAM* de la placa principal.

Para conocer los K, teclee:

PRINT INT(FRE(0)/1024)

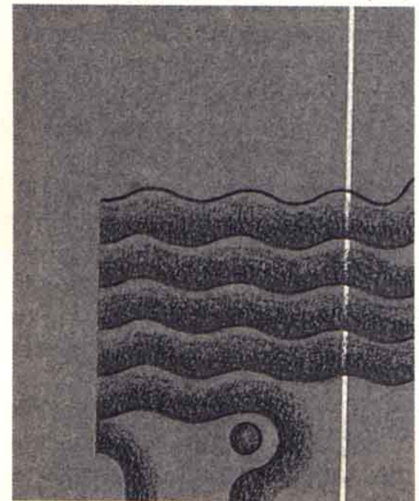
El resultado son los K disponibles. En algunos ordenadores este número ya viene en el anagrama de presentación. Como las respuestas pueden ser variadas diremos que desde 64 hasta 16 son todas correctas. Entonces:

¿El ordenador dispone de 64 K y yo sólo dispongo de 32? Pues sí. Quizá menos si dispone de disquette. El sistema necesita de esa mitad para sus rutinas internas



con las que interpretar el *BASIC* y facilitarnos la labor. Esta memoria es la que se llama *ROM*. Podemos leerla pero no modificarla.

Con este dato aclaramos un paso en el progreso de comprender la capacidad de la memoria. Dis-



ponemos de n *Kilobytes* de *RAM*.

Si pensamos que un programa de unos 20 K ya se puede considerar amplio; está claro que una memoria de 32/48 K colmará nuestras necesidades, pero sólo a primera vista, pues depende de lo que pretendamos. Para dar una idea pensemos que un escrito de unos 20/40 folios, doble espacio a máquina; 6/10 dibujos en otras tantas pantallas completas ocupan la extensión citada de esos 32/40 K aproximadamente. Dejamos 8 de reserva.

Alguien puede pensar que 48 K para juegos son suficientes y tendrá razón, pero siempre que tengamos en cuenta que los juegos comprados suelen venir en código máquina y profundamente estudiados por profesionales para ahorrar memoria.

Un programa depurado escrito en *BASIC* de la misma estructura que el citado puede ocuparnos el doble de memoria y en ese caso nos quedaría corta la que disponemos en principio.

Pero también se puede ahorrar memoria cuando convenga.

Veamos:

CLS y/o NEW HOME en MSX

(A fuerza de resultar insistentes repetimos que cada vez que terminemos una o varias instrucciones debemos actuar *ENTER, RETURN* o *CR)*

```
10 PRINT ▲"PRUEBA DE ME-
     MORIA": PRINT
20 PRINT  "E▲S▲P▲A▲C▲I▲O▲S"
30 END                      (1)
```

Actuar ahora:

PRINT FRE(Ø)

y anotar el resultado.

Volver a escribir el programa 1Ø/3Ø suprimiendo los espacios (▲).



Actuar de nuevo:

PRINT FRE(Ø)

y restarla del resultado anterior, lo cual quiere decir que se ahorran alrededor de los 1Ø *bytes* con la segunda versión del programa.

Sin embargo en programas largos y después de un cierto tiempo, hemos encontrado los siguientes aspectos negativos:

CONFUSION
OSCURIDAD
FALTA DE NOTAS ACLARATIVAS
DIFICULTAD DE RESERVA (DIM)

Los profesionales deben reducir memoria en aras de la economía, pero los aficionados, sobre todo en la época de aprendizaje, no podemos realizar los programas demasiado apelmazados y carentes de indicaciones *(REM)*.

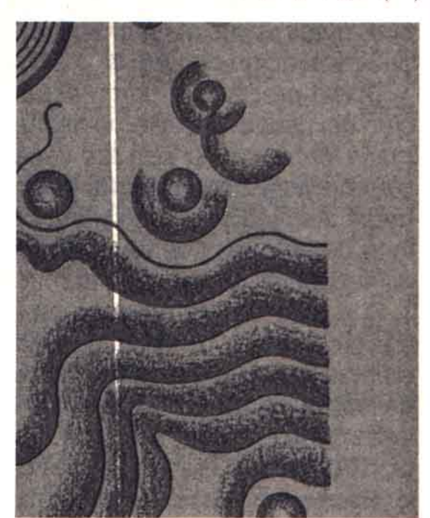
El mejor método para conocer exactamente la memoria de que dispondremos nos los «dirá» el propio ordenador de modo explícito. Esto sería muy aconsejable antes de tomar una decisión en la adquisición de un modelo determinado.

Reservas de memoria

Si pensamos introducir en la memoria datos que dependan de letras, como nombres, direcciones, diagramas; o también expresiones de letras con números tales como números de teléfonos, listas de precios, etc. que son lo que llamamos variables alfanuméricas. Hemos de saber la reserva de que dispone nuestro ordenador y si vamos a sobrepasar dicha reserva.

Esta clase de lugar que ocupa la reserva se define también en *bytes* y se logra averiguar su cantidad así:

PRINT FRE ("")



y *ENTER*. Aparecerá un número entre 1ØØ y 4ØØ aproximadamente ¿son suficientes? si la res-

(1) ▲ representa un espacio en blanco; « » para variables alfanuméricas.

puesta fuese negativa tenemos la solución inmediata:

CLEAR 1ØØØ



y *ENTER* (Menuda pesadilla). Si volvemos a investigar ahora la capacidad de sólo la memoria reservada *(PRINT FRE* ("")) observaremos el aún fresco aumento de una zona (i8ØØ *bytes!)* en detrimento de la memoria general, es decir, actuando:

PRINT FRE (Ø)

el resultado ahora de la memoria *RAM* disponible será inferior en 8ØØ *bytes* al que obtuvimos antes de la actuación del *CLEAR* 1ØØØ.

Como previsión y como actuación del usuario, la reserva de memoria debe contar con las variables previstas. La memoria global habrá disminuido y es muy importante tenerlo en cuenta antes de confeccionar los programas.

Si a todo esto le agregamos el posible empleo de grupos o elementos (cartas de la baraja, fichas de juegos como el dominó o conjuntos de componentes con relación de distintos elementos; por citar unos ejemplos) debemos establecer de antemano la «dimensión» de cada grupo de variables suponiendo que su capacidad sea superior a 1Ø, cantidad que es la común en la mayoría de los casos a nuestros ordenadores, del modo siguiente:

DIM A(1ØØØ)

ejecución en modo directo y después y a continuación el consabido:

PRINT FRE(Ø)

que nos permitirá ver con sorpresa la fuga de 8K, expresión que significa exactamente la reserva de 8 *bytes* por cada uno de los datos DIMensionados (8x1ØØØ). Si alguno espera todavía al *DIM* A(1ØØØ) o al *PRINT FRE*(0) lo recuerdo por enésima vez que actúe *ENTER* después de cada instrucción o línea completa (¡¡VALE!!).



Si os parece exagerado el número mil para reserva de datos bajarlo y probad a vuestro gusto. Un poco de osadía e imaginación son las virtudes de todo microadicto que unidas a la paciencia forman la estructura básica del profesional.

A cada inconveniente le vamos encontramos algún subterfugio capaz de paliarlo. Borrad la pantalla *(NEW* y *CLS)* y anotad:

DIM A!(1ØØØ)

algo ha cambiado en la reserva de memoria. Ahora son sólo 4 *kilobytes* ya que con el símbolo ! le hemos indicado que no nos es necesaria una gran precisión en los números, reducción de decimales que nos permite no bajar el listón de la memoria demasiado. Y si no necesitamos decimal alguno:

DIM A%(1ØØØ)

nuevo símolo que indica para todos los elementos de A una forma entera. Ahora sólo necesitaremos 2K de reserva aproximadamente. Debemos anular cada versión de A ya que el ordenador interpreta:

A! A% A

como tres variables totalmente distintas, lo cual aprovecharemos más adelante para aumentar el campo alfabético de las variables.

Conclusión

Hemos iniciado los primeros pasos de la memoria *RAM* con las facetas que suplen las carencias estructurales, *ENTER* se nos ha repetido hasta la saciedad (Perdón), en los próximos artículos veremos por medio de programas breves



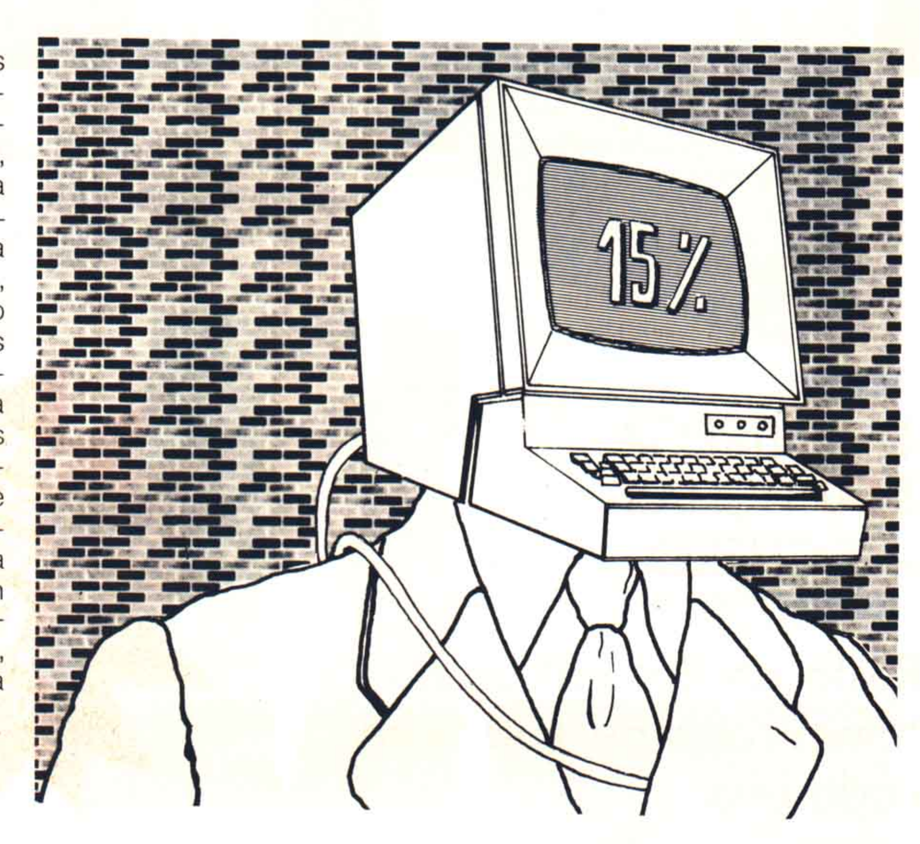
las posibilidades de reducir memoria y tiempo.

José Leal Rodríguez

Amortización de préstamos

Las Matemáticas Financieras suele ser el coco de muchos estudiantes de Ciencias Empresariales. La monotonía de los cálculos, así como el tiempo que se dedica a cada problema objeto de estudio, acaba por hacer mella y a crear situaciones problemáticas, sobre todo cuando se está a punto de obtener la solución. Muchos errores no vienen por desconocimiento de la materia, sino por esa especie de «virus» que afecta a los dedos, y que hace que se cometan equivocaciones, cuando se están efectuando cálculos a mano. Esto tiene solución. Llevar una calculadora programable con un programa que permita resolver dichos problemas. En nuestro caso, no podemos poner la calculadora pero sí el programa.

Carlos García Fernández
Vizcaya




```
10 REM ********************
20 REM **  PRESENTACION  **
30 REM ********************
40 REM
50 CLS:KEY OFF:WIDTH 40:COLOR 15,1
60 FOR K=1 TO 10
70 LOCATE 7,22:PRINT"AMORTIZACION DE
    PRESTAMOS"
80 FOR H=1 TO 100:NEXT H
90 CLS:FOR H=1 TO 100:NEXT H
100 NEXT K
110 LOCATE 7,22:PRINT"AMORTIZACION DE
    PRESTAMOS"
120 FOR K=1 TO 500:NEXT K
130 FOR K=1 TO 18:PRINT:NEXT K
140 LOCATE 0,7
150 PRINT"    1 - SISTEMA FRANCES."
160 PRINT"    2 - SISTEMA ALEMAN."
170 PRINT"    3 - SISTEMA UNIFORME."
180 PRINT"    4 - SISTEMA AMERICANO."
190 PRINT:PRINT:PRINT"      PULSA OPCIO
    N (1-4)."
200 Z$=INPUT$(1)
210 Z=VAL(Z$):IF Z>4 OR Z<1 THEN 190
220 ON Z GOSUB 1000,2000,3000,4000
230 CLS:PRINT:PRINT
240 INPUT" DESEAS HACER OTRO CUADRO";
    N$
250 N$=LEFT$(N$,1)
260 IF N$="S" OR N$="s" THEN RUN
```

```
270 END
280 REM
1000 REM  ********************
1010 REM  **  SIST. FRANCES  **
1020 REM  ********************
1030 REM
1040 CLS:PRINT
1050 INPUT"ESCRIBE EL CAPITAL PRESTAD
     O.              CAP.";CC
1060 INPUT"ESCRIBE EL Nº DE AÑOS QUE
     DURA LA OP.    N.";N
1070 INPUT"ESCRIBE EL TIPO DE INTERES
     .              T/I.";TI
1080 IF TI>1 THEN TI=TI/100
1090 CV(0)=CC
1100 FOR K=1 TO N
1110 A(K)=CC*((1+TI)^N*TI)/((1+TI)^N-
     1)
1120 I(K)=CV(K-1)*TI
1130 CA(K)=A(K)-I(K)
1140 E(K)=E(K-1)+CA(K)
1150 CV(K)=CC-E(K)
1160 NEXT K:SIST=0
1170 GOSUB 5000:RETURN
1180 REM
1190 REM
2000 REM  *********************
2010 REM  **  SISTEMA ALEMAN  **
2020 REM  *********************
2030 REM
2040 CLS:PRINT
2050 INPUT"ESCRIBE EL CAPITAL PRESTAD
     O.              CAP.";CC
2060 INPUT"ESCRIBE EL Nº DE AÑOS QUE
     DURA LA OP.    N.";N
2070 INPUT"ESCRIBE EL TIPO DE INTERES
     .              T/I.";TI
2080 IF TI>1 THEN TI=TI/100
2090 CV(0)=CC:I(0)=CC*TI
2100 FOR K=1 TO N
2110 A(K)=(CC*TI)/(1-(1-TI)^N)
2120 NEXT K:CA(N)=A(N)
2130 FOR K=N-1 TO 1 STEP -1
2140 CA(K)=CA(K+1)*(1-TI)
2150 NEXT K
2160 FOR K=1 TO N
2170 E(K)=E(K-1)+CA(K)
2180 CV(K)=CC-E(K)
2190 I(K)=CV(K)*TI
2200 NEXT K:SIST=0
2210 GOSUB 5000:RETURN
2220 REM
2230 REM
3000 REM  ***********************
3010 REM  **  SISTEMA UNIFORME  **
3020 REM  ***********************
3030 REM
3040 CLS:PRINT
3050 INPUT"ESCRIBE EL CAPITAL PRESTAD
     O.              CAP";CC
3060 INPUT"ESCRIBE EL Nº DE AÑOS QUE
     DURA LA OP.    N";N
3070 INPUT"ESCRIBE EL TIPO DE INTERES
     .              T/I";TI
3080 IF TI>1 THEN TI=TI/100
3090 CV(0)=CC
3100 FOR K=1 TO N
3110 CA(K)=CC/N
3120 E(K)=E(K-1)+CA(K)
3130 CV(K)=CC-E(K)
3140 I(K)=CV(K)*TI
3150 A(K)=I(K)+CA(K)
3160 NEXT K:SIST=0
3170 GOSUB 5000:RETURN
3180 REM
3190 REM
4000 REM  *************************
4010 REM  **  SISTEMA AMERICANO  **
4020 REM  *************************
4030 REM
4040 CLS:PRINT
4050 INPUT"ESCRIBE EL CAPITAL PRESTAD
     O.              CAP";CC
4060 INPUT"ESCRIBE EL Nº DE AÑOS QUE
     DURA LA OP.    N";N
4070 INPUT"ESCRIBE EL TIPO DE INTERES
      DE LA OP.     T/I";TI
4080 INPUT"ESCRIBE EL TIPO DE INTERES
      DEL MERCADO  T/I MERC";TM
4090 IF TI>1 THEN TI=TI/100
4100 IF TM>1 THEN TM=TM/100
4105 DP(0)=CC
4110 FOR K=1 TO N
4120 I(K)=CC*TI
4130 CA(K)=CC*TM/((1+TM)^N-1)
4140 A(K)=CA(K)+I(K)
4150 FA(K)=CA(K)+FA(K-1)*(1+TM)
4160 FI(K)=FA(K-1)*TM
4170 DP(K)=CC-FA(K)
4180 NEXT K
4190 SIST=4
4200 GOSUB 5000:RETURN
4210 REM
4220 REM
5000 REM ***************************
     *
5010 REM **  PRESENTACION DE RDOS.  *
     *
5020 REM ***************************
     *
5030 REM
5040 CLS:PRINT
5050 PRINT" ** ANUALIDADES **":PRINT
5060 FOR K=0 TO N
5070 PRINT"AÑO=";K;"  =>";INT(A(K))
5080 NEXT K
5090 GOSUB 9000
5100 PRINT" ** CUOTAS DE AMORTIZACION
      **":PRINT
5110 FOR K=0 TO N
5120 PRINT"AÑO=";K;"  =>";INT(CA(K))
5130 NEXT K
```

```
5140 GOSUB 9000
5150 PRINT" ** CUOTAS DE INTERES  **"
     :PRINT
5160 FOR K=0 TO N
5170 PRINT"AÑO=";K;"  =>";INT(I(K))
5180 NEXT K
5190 GOSUB 9000
5200 IF SIST=4 THEN 5320
5210 PRINT" ** CAPITAL AMORTIZADO  **
     ":PRINT
5220 FOR K=0 TO N
5230 PRINT"AÑO=";K;"  =>";INT(E(K))
5240 NEXT K
5250 GOSUB 9000
5260 PRINT" ** CAPITAL VIVO  **":PRIN
     T
5270 FOR K=0 TO N
5280 PRINT"AÑO=";K;"  =>";INT(CV(K))
5290 NEXT K
5295 GOSUB 9000
5300 GOSUB 6000
5310 RETURN
5320 PRINT" ** FONDO DE AMORTIZACION
      **":PRINT
5330 FOR K=0 TO N
5340 PRINT"AÑO=";K;"  =>";INT(FA(K))
5350 NEXT K
5360 GOSUB 9000
5370 PRINT" ** INTERES DEL FONDO  **"
     :PRINT
5380 FOR K=0 TO N
5390 PRINT"AÑO=";K;"  =>";INT(FI(K))
5400 NEXT K
5410 GOSUB 9000
5420 PRINT" ** DEUDA PENDIENTE  **":P
     RINT
5430 FOR K=0 TO N
5440 PRINT"AÑO=";K;"  =>";INT(DP(K))
5450 NEXT K
5460 GOSUB 9000
5470 GOSUB 6000
5490 RETURN
6000 PRINT:PRINT
6010 INPUT" DESEAS VER OTRA VEZ LOS R
     ESULTADOS";N$:N$=LEFT$(N$,1)
6020 IF N$="S" OR N$="s" THEN 5000
9000 PRINT:PRINT
9010 PRINT"PULSA UNA TECLA"
9020 IF INKEY$="" THEN 9020
9030 CLS:PRINT:RETURN
```



Centrado de textos

La rutina siguiente, permite unas buenas aplicaciones en todo tipo de programas, ya sean de gestión, aplicación o juegos.

La variable TX$ contiene el texto a centrar, mientras que la variable Y almacena la coordenada y.

Al igual que la rutina anterior, se puede utilizar dentro de un programa, siempre y cuando se realicen las modificaciones oportunas.

```
10 REM ###################
20 REM Centrado de textos
30 REM ###################
40 A$="ESTO ES UNA PRUEBA DE CENTRADO"
50 SCREEN 0
60 X=(37-LEN(A$))/2
70 LOCATE X,Y:PRINT A$
```

LOS REGISTROS DEL VDP

RG	BIT 7	6	5	4	3	2	1	0
0	NO USADOS						SCREEN 1	VIDEO EXTERNO
1	SELECCIONA RAM	ACTIVA PANTALL	ACTIVA INTRRUP	SCREEN 0	SCREEN 2	EN RESERVA	BITS SPRITE	AUMENTO SPRITE
2	NO USADOS				DIRECCION DE LA TABLA DE DENOMINACIONES = REG2 * 400 hex			
3	DIRECCION DE LA TABLA DE COLOR = REGISTRO 3 * 40 hex							
4	NO USADOS					DIR. TABLA GENERACION DE PATRONES= REG4 * 800 hex		
5	NO USADO	DIRECCION DE LA TABLA DE ATRIBUTOS DE SPRITES = REGISTRO 5 * 80 hex						
6	NO USADOS					DIR. GENERACION PATRONES DE SPRITES=REG6 * 800 hex		
7	COLOR TEXTO				COLOR FONDO			
8	BANDERA INTRRUP	BANDERA 5SPRTES	CHOQUE SPRITES	NUMERO DE PLANO DEL QUINTO SPRITE				

Aclaración

Los meses de verano, son bien conocidos por los despistes que la época estival produce, sin duda, por el agobiante calor, las vacaciones, etc. Además, teniendo en cuenta que Busy cuando ataca, lo hace de verdad, no le resultará extraño al lector que en el artículo del SVI-318/328, publicado en el número de julio/agosto, faltara una importante tabla que lo acompañaba. Dicha tabla hace referencia a los registros VDP.

Lectura de la cabecera del cassette

Esta corta rutina, es muy útil a la hora de pasar los ficheros binarios (programas en código máquina) de cinta a disco. La rutina lee la cabecera de estos ficheros y proporciona todos los datos necesarios sobre el programa en cuestión, tales como: nombre, dirección de comienzo, de final y autoejecución.

```
10 REM lectura de la cabecera del casette
20 REM
30 REM
40 CLEAR 200,&HBFFF
50 CLS
60 PRINT "lectura de la cabecera del case
    tte"
70 PRINT:PRINT:PRINT
80 FOR I=&H9100 TO &H912B
90 READ A$
100 POKE I,VAL("&h"+A$)
110 NEXT I
120 DEFUSR=&H9100
130 DATA f3,0e,d0,cd,e9,72,06,0a,cd,d4,72
    ,b9,20
140 DATA f5,10,f8,21,00,90,06,06,cd,d4,72
    ,77,23
150 DATA 10,f9,cd,e9,72,06,06,cd,d4,72,77
    ,23,10,f9,cd,e7,00,c9
160 FOR I=&H9000 TO &H9020:POKE I,0:NEXT
170 L=USR(0)
180 N$=""
190 FOR I=&H9000 TO &H9005
200 N$=N$+CHR$(PEEK(I))
210 NEXT I
220 PRINT "NOMBRE ";N$
230 PRINT
240 I=&H9006
250 AD$=HEX$(PEEK(I)+256*PEEK(I+1))
260 PRINT "DIRECCION DE COMIENZO ";AD$
270 PRINT
280 AF$=HEX$(PEEK(I+2)+256*PEEK(I+3))
290 PRINT "DIRECCION FINAL ";AF$
300 PRINT
310 PE$=HEX$(PEEK(I+4)+256*PEEK(I+5))
320 PRINT "DIRECCION DE EJECUCION ";PE$
330 PRINT
```

Rincón del lector

INFORMACION ADICIONAL SOBRE MSX

¿Me podrían informar dónde puedo encontrar un mapa de memoria del MSX, muy detallado, así como un mapa de las variables del sistema y las direcciones de la ROM más importantes y cómo se utilizan?

Juan Carlos Cillernelos
Burgos

Dentro de las diferentes publicaciones existentes en el mercado, sólo hay que merezca especial atención; «MSX. Guía del programador y manual de referencia». Este libro, además de orientar al usuario en todos los aspectos del BASIC MSX, le ayuda a comprender las características internas más importantes.

La última parte de este libro contiene, desde un capítulo dedicado a las instrucciones RST hasta las llamadas a BIOS. Es el manual más completo que hemos visto.

PROGRAMAR LA TARJETA INTELIGENTE

¿Es posible programar la tarjeta Bee-Card a través del bus de expansión o de cartucho de un ordenador MSX? En cualquier caso, ¿cómo?

José Ruiz Hierrezuelo
Granada

En efecto, es posible programar la tarjeta, pero para ello es necesario disponer del cartucho adecuado. En Inglaterra, venden un kit para programar cartuchos, pero hasta el momento no lo hemos visto funcionar.



CONECTOR DE JOYSTICK

Poseo un joystick de tipo analógico de potenciómetro y un conector tipo Delta para conectarlo a mi MSX, pero desconozco que pines se conectan a los tres puntos del potenciómetro y también me gustaría que me confirmaran la instrucción necesaria para leer el valor del port.

Cayetano Gómez
Sevilla

En la figura adjunta, se detallan las patillas de conexión del port de joystick, así como sus correspondientes funciones.

Con respecto a la otra pregunta las órdenes relacionadas con el port de joystick son PAD y PDL.

PROGRAMAS QUE NO FUNCIONAN

He copiado alguno de sus programas de juegos, correctamente y no funcionan. ¿Están puestos sólo como ejemplos o realmente deben salir en pantalla?

También me han dicho que hay un bloqueo en ellos, ¿es cierto?

José Puig Sánchez
Barcelona

No nos dices nada sobre qué programa has copiado, ni el tipo de error que aparece, ya que los programas están puestos para fomentar el envío de vuestras colaboraciones y vuestras ideas y sólo se publican las que funcionan debidamente. Además, los programas que se reciben son comprobados, no en un ordenador, sino en varios que tenemos para, de esta forma, asegurarnos que son correctos.

Nº PIN	Señal
1	FWD
2	BACK
3	LEFT
4	RIGHT
5	+5V
6	TRG 1
7	TRG 2
8	Output
9	GND

Diagrama de conexión

1 2 3 4 5
6 7 8 9